Ein Konzept zur Überwachung und Mißbrauchserkennung bei Grid-Proxy-Credentials

Von der Fakultät für Elektrotechnik und Informatik der
Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor-Ingenieur
(Dr. ing.)

genehmigte Dissertation

von
Herr M. Sc. Christopher Kunz
geboren am 22.10.1979 in Gütersloh

2011

Referent: Prof. Matthew Smith

Koreferent: Prof. Gabriele von Voigt

Koreferent: Prof. Bernd Freisleben

Tag der Promotion: 13.12.2011

Vorwort

Die vorliegende Arbeit entstand im Rahmen des BMBF-Projektes *D-Grid*, während der Tätigkeit als wissenschaftlicher Mitarbeiter am Regionalen Rechenzentrum für Niedersachsen und Forschungszentrum L3S.

Ich danke ganz besonders Herrn Prof. Smith für die Unterstützung bei der Erstellung dieser Arbeit sowie für das entgegengebrachte Vertrauen und wertvolle Denkanstöße.

Herrn Prof. Dr. Freisleben und Frau Prof. Dr. von Voigt danke ich für die Übernahme der Korreferate.

Besonderer Dank gebührt auch meinen Kollegen, die in intensiven Diskussionen und Gesprächen geholfen haben, Gedankengänge zu vertiefen und konkretisieren, sowie ihre vielfältige Unterstützung bei Publikationen im Rahmen dieser Dissertation eingebracht haben. Hier möchte ich besonders Christian Szongott danken, dessen Implementation sehr wertvoll für den weiteren Verlauf dieser Arbeit war, sowie Nina Tahmasebi vom L3S für ihre Mitarbeit an Grid-Simulation und Missbrauchserkennung.

Ich möchte mich bei den Kollegen der Filoo GmbH dafür bedanken, dass sie mir unter großem persönlichen Einsatz den Rücken freigehalten haben, sowie bei meinem Freundeskreis, besonders Joana Santos, für die moralische Unterstützung.

Meinen Eltern, die mich stets ermutigt und meine Neugier angeregt haben, danke ich ebenso für ihre Unterstützung wie meiner Freundin Miriam Reichelt, deren Anmerkungen zu Stil und Ortographie diese Arbeit deutlich verbessert haben.

Zusammenfassung

Die vorliegende Dissertation präsentiert ein Konzept zur Auditierung und Mißbrauchserkennung in verteilten Umgebungen. Besonderes Augenmerk wird hierbei auf Grid-Computing-Umgebungen gelegt.

Das grundlegende Paradigma des Grid-Computing sieht vor, daß Nutzer Aufgaben an eine verteilte Infrastruktur abgeben können, ohne während der Ausführung dieser Aufgaben mit den ausführenden Systemen interagieren zu müssen. Eventuell notwendige Authentifizierungsvorgänge, etwa bei Datenspeichern, werden durch Delegation von Rechten und Mechanismen zum Single-Sign-On durchgeführt. Gleichsam wird dem Nutzer jedoch die Kontrolle über seine Authentifizierungs-Tokens entzogen, was die Mißbrauchserkennung sehr erschwert.

In der vorliegenden Dissertation wird ein Konzept für eine grid-weite Auditing-Infrastruktur zum Auditing und zur Mißbrauchserkennung von Grid-Authentifizierungs-Credentials vorgestellt. Diese Infrastruktur wird auf Basis des Globus Toolkit entwickelt und nutzt Standards und Protokolle, die von diesem führenden Grid-Toolkit vorgegeben werden. Ziel dieser Arbeit ist es, die Erkennung mißbräuchlicher Nutzung von Authentifizierungsdaten im Grid zu ermöglichen, indem jede Nutzung dieser Daten aufgezeichnet und gespeichert wird.

Für diese Dissertation wurde das Globus Toolkit modifiziert, um die in Proxy-Credentials integrierten Zusatzinformationen auszuwerten, sobald ein solches Credential zur Authentifizierung verwendet wird. Ein Audit-Datensatz wird dann an eine dezentrale Serverinfrastruktur gesendet, die diese Informationen sammelt und dem Endnutzer aufbereitet zur Verfügung stellt.

Desweiteren wird in dieser Arbeit ein Ansatz zur Auswertung der Audit-Datensätze vorgestellt, der mithilfe bayesscher Klassifikatoren die Rohdaten in eine für den Nutzer nutzbare Form bringt und ihn so in die Lage versetzt, zuverlässig beurteilen zu können, ob seine Authentifizierungsdaten mißbraucht wurden.

Grid-Computing, Auditierung, Mißbrauchserkennung

Abstract

This thesis presents a concept for abuse detection and auditing in distributed computing environments. It gives special attention to Grid Computing and its standardized authentication mechanisms.

The paradigm of Grid Computing mandates that users who wish to compute a given task need not survey its execution in the Grid; mechanisms for Single-Sign-on and delegation of rights remove interactivity that might otherwise be necessary when using a distributed computing environment. However, with the concept of delegation of rights, control over the user's login credentials is also removed from them, making abuse detection difficult.

This thesis aims to tackle that issue by presenting a concept and implementation for a Grid-wide infrastructure for credential auditing and abuse detection. This infrastructure is implemented on top of the Globus Toolkit, using standards and protocols defined by this leading Grid toolkit. Its central goal is to track how a user's authentication token ("proxy certificate") is used and transported in the Grid. To achieve this goal, the concept presented in this thesis takes advantage of the fact that users in a Grid environment can embed custom information in their authentication tokens before passing these into the Grid as part of a job workflow.

As a part of this thesis, the Grid middleware "Globus Toolkit" was modified to evaluate these pieces of custom information whenever an authentication token is used during the course of a Grid job. A usage record is then sent to a server infrastructure which, in turn, aggregates usage records and presents them to the user.

Apart from this means of surveying and storing Grid usage information, this thesis presents an approach at evaluating usage records using Bayesian classifiers and thus converting the raw data into a more meaningful presentation for end users. These will then be able to estimate if their authentication tokens were misused.

Grid Computing, Auditing, Abuse detection

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

AAI	Authentifizierungs- und Autorisierungs-Infrastruktur
ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
ARC	Advanced Resource Connector
AuthN	Authentifizierung
AuthZ	Autorisierung
BDSG	Bundesdatenschutzgesetz
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CAS	Community Authorization Service
CE	Computing Element
CERN	Centre Européenne pour la Recherche Nucléaire
CERT	Computer Emergency Response Team
DAG	Directed Acyclic Graph
DAO	Data Access Object
dDoS	distributed Denial of Service
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DFN	Deutsches Forschungs-Netz
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name
DoS	Denial of Service
EEC	End-Entity Certificate
EGEE	Enabling Grids for E-sciencE
EGI	European Grid Initiative

EUGridPMA	European Union Grid Policy Management Authority
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GRAM	Grid Resource Allocation and Management
GSI	Grid Security Infrastructure
GT4	Globus Toolkit 4
HIDS	Host-based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol, Secure
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
ISMS	Information Security Management System
ISO	International Standardization Organization
ITU-T	International Telegraphic Union, Standardization Sector
JSDL	Job Submission Description Language
KIT	Karlsruhe Institute of Technology
LCG	LHC Computing Grid
LDAP	Lightweight Directory Access Protocol
LHC	Large Hadron Collider
NIDS	Network Intrusion Detection System
OCSP	Online Certificate Status Protocol
OGF	Open Grid Forum
OGSA	Open Grid Services Architecture
OGSA-DAI	Open Grid Service Architecture - Data Access & Integration
OID	Object Identifier
PHP	PHP Hypertext Preprocessor
PKI	Public-Key Infrastructure
RDBMS	Relational Database Management System
RFT	Reliable File Transfer
RSA	'Rivest, Shamir, Adelman'; nach seinen Erfindern benannter Algorithmus
RSL	Resource Specification Language
SHA	Secure Hash Algorithm
SQL	Structured Query Language

SSH	Secure Shell
SSL	Secure Sockets Layer
SSO	Single Sign On
TAGPMA	The Americas Grid Policy Management Authority
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UI	User Interface
UNICORE	Uniform Interface to Computing Resources
URL	Uniform Resource Locator
VO	Virtuelle Organisation
VOMRS	VO Management & Registration Service
VOMS	VO Management Service
WEKA	Waikato Environment for Knowledge Analysis
WMS	Workload Management System
WS	Web Service
WS-GRAM	Web Service Grid Resource Allocation and Management
WSRF	Web Service Resource Framework

Kapitel 1

Einführung

Eine Einleitung in das „Grid Computing“ sowie in die Motivation und Ausgangssituation dieser Arbeit wird im folgenden Kapitel vorgestellt. Es gibt ebenfalls Auskunft über Aufbau und Gliederung der vorliegenden Dissertation.

Eine Einordnung in den wissenschaftlich-technischen Kontext wird vorgenommen, um die Berührungs- und Differenzierungspunkte dieser Arbeit mit ähnlichen und verwandten Arbeiten herauszuarbeiten.

1.1 Einleitung

Wissenschaftliche Anwendungen waren und sind ein wesentlicher Motor in der Entwicklung verteilter Infrastrukturen. Das Internet, ursprünglich im Rahmen eines Militärprojekts entwickelt, wurde lange Zeit hauptsächlich von Wissenschaftlern zur Kommunikation untereinander genutzt; einer breiten Bevölkerung steht es erst seit etwa zwanzig Jahren zur Verfügung. Während Speicher- und Übertragungskapazitäten in Nah- und Weitverkehrsnetzen in den vergangenen Jahrzehnten exponentiell gestiegen sind, hat sich die Komplexität der im Forschungsumfeld zu berechnenden Probleme gleichermaßen, oftmals sogar stärker erhöht. In praktisch jeder wissenschaftlichen Disziplin werden komplexe Algorithmen auf große, experimentell gewonnene Datenmengen angewandt, um aus diesen Rohdaten Erkenntnisse zu gewinnen.

Da viele Forschungsinstitutionen über eigene Rechenanlagen verfügen, liegt es nahe, die Koppelung dieser Cluster zu einer massiven verteilten Infrastruktur anzustreben, deren Rechenkapazität sogar für die komplexesten zu lösenden Probleme ausreicht. Um es den Wissenschaftlern aus informatikfernen Disziplinen zu erleichtern, eine solche verteilte Infrastruktur zu nutzen, wird eine gemeinsame Nutzerschnittstelle und Middleware angestrebt. Ian Foster, Vordenker der von ihm als „Grid Computing“ getauften Vision, skizzierte diesen Gedanken wie folgt:

> „A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.“
>
> (Ian Foster in [FK03])

Grid Computing findet mittlerweile in verschiedenen wissenschaftlichen Disziplinen Anwendung und stellt Forscher sowie Administratoren vor neue Herausforderungen. Besondere Beachtung verdienen hier die zu lösenden Probleme der Sicherheit, also insbesondere der Vertraulichkeit, Authentizität, Integrität und Verfügbarkeit von Daten und Diensten. Eine komplexe Grid-Sicherheits-Infrastruktur löst viele der offenen Fragen mithilfe digitaler Zertifikate, ist dabei für viele Nutzer jedoch wenig transparent. Mit einem Konzept zum Auditing von Grid-Credentials soll dieser mangelnden Transparenz Abhilfe geschaffen und damit die Sicherheit im Grid erhöht werden.

1.2 Motivation

Um die zentrale Anforderung des Grid-Computing, nämlich die autarke Abarbeitung von komplexen Rechenvorgängen durch eine dezentrale Infrastruktur, umsetzen zu können, wurden neue Authentifizierungsmechanismen notwendig.

Die in herkömmlichen verteilten Infrastrukturen noch immer übliche Authentifizierung mittels einer nur dem Nutzer bekannten Information („Knowledge Factor Authentication“) kann zwar in einer Grid-Infrastruktur prinzipiell eingesetzt werden (etwa bei der Abgabe einer Rechenaufgabe), wird jedoch bei jeder Teil- oder Folgeaufgabe erneut notwendig. Da ex ante nicht feststellbar ist, ob und zu welchem Zeitpunkt solche Authentifizierungsvorgänge notwendig werden könnten, wäre eine ständige Überwachung des Grid-Jobs durch den Nutzer notwendig und die Nutzbarkeit von Grids deutlich eingeschränkt.

In Grid-Infrastrukturen wird die Authentifizierung des Nutzers (oder durch ihn beauftragter Instanzen) vorgenommen, indem der Besitz eines Sicherheitsmerkmals durch die zu authentifizierende Partei nachgewiesen wird. Dieses Merkmal ist der private Schlüssel, der zum öffentlichen Schlüssel im X.509-Zertifikat des Nutzers passt. Durch die Nutzung von „Proxy-Zertifikaten“ – vom Nutzer selbst signierten Ableitungen des eigentlichen Nutzerzertifikats – können Grid-Ressourcen auf eine sichere Art und Weise mit temporär gültigen Sicherheitsmerkmalen zur Ein-Faktor-Authentifizierung ausgestattet werden. Da die kryptographischen Signaturen jedes Zertifikats mithilfe des ausstellenden Zertifikats geprüft werden können, ergibt sich eine Vertrauenskette vom Proxy zum Nutzerzertifikat – Grid-Ressourcen können sich so als „vom Nutzer beauftragt“ ausweisen.

Die Nutzung dieses Verfahrens wirft neue Sicherheitsfragen auf. So ist es, etwa unter Ausnutzung einer Sicherheitslücke einer Grid-Komponente, möglich, in den Besitz gültiger Proxy-Credentials (also von Proxy-Zertifikaten und den zugehörigen privaten Schlüsseln) zu gelangen und somit über dasselbe Sicherheitsmerkmal zu verfügen wie eine legitim beauftragte Grid-Ressource. Gelangt ein unautorisierter Dritter in den Besitz von gültigen Proxy-Credentials, so kann dieser – im Rahmen der Berechtigungen des Nutzers, dessen Credentials er benutzt – auf Daten lesend und schreibend zugreifen, bereits laufende Grid-Jobs manipulieren und neue Jobs submittieren. Obgleich die Möglichkeit unautorisierter Job-Submission empfindliche Folgen nach sich ziehen kann, ist die Manipulation oder Zerstörung von im Grid

gelagerten Daten ein ungleich erschreckenderes Missbrauchsszenario.

Der Verlust oder die Manipulation von Forschungs- und Experimentaldaten, die teilweise nur durch zeit- und kostenintensive Wiederholung der zugrundeliegenden Experimente rekonstruiert werden können, hätte immense Auswirkungen auf die betroffenen Nutzer. Bei Anwendern mit hohem Schutzbedürfnis, etwa in der medizinischen Datenverarbeitung, geraten gar höchstpersönliche Datensätze in Gefahr.

Der Nutzer hat in herkömmlichen Grid-Infrastrukturen keine Möglichkeit, Missbrauch seiner Credentials zu erkennen, da diese im Grid ohne Nutzerinteraktion erstellt und verteilt werden. Zudem fehlen automatische Erkennungsmöglichkeiten für Credential-Missbrauch; eine manuelle Überprüfung ist aufgrund der schieren Masse stichprobenartig möglich. So ist es dem Nutzer meist nicht möglich, widerrechtliche Nutzung seiner Credentials zu erkennen. Aus dieser fehlenden Funktionalität entstehen im Missbrauchsfall schwerwiegende Sicherheitsimplikationen.

Durch die in Grid-Infrastrukturen systemimmanente Notwendigkeit, zusammen mit einer Teilmenge der eigenen Rechte auch ein Authentifizierungs-Token an eine nicht nachvollziehbare Menge von Entitäten weiterzugeben, fällt es Nutzern aus dem wissenschaftlichen und kommerziellen Umfeld schwer, Vertrauen in die Grid-Nutzung aufzubauen.

1.3 Zielsetzung

Wie im vorangegangenen Abschnitt (und in Kapitel 3) ausgeführt, ist die Infrastruktur zur Authentifizierung und Autorisierung in Grids derzeit für den Nutzer, aber auch für den Ressourcenbetreiber prinzipbedingt intransparent. Aus diesem Umstand leitet sich die grundlegende Zielsetzung dieser Arbeit ab, mehr Transparenz bei der Delegation von Authentifizierungsinformationen zu schaffen und somit beim Nutzer eine Vertrauensbasis herzustellen. Zu diesem Zweck wurde ein Konzept zum Auditing von Proxy-Credentials entworfen und implementiert. Mittels einer Modifikation der Security-Infrastrukturbibliotheken in der Grid-Middleware wird eine Rückmeldung an einen Auditingdienst ausgelöst, sobald ein Proxy-Credential eingesetzt wird.

Der Endnutzer erhält durch in einem Webportal aggregierte Informationen eine Möglichkeit, den Weg nachzuvollziehen, den das von ihm erzeugte Credential im Grid genommen hat; erfahrene Benutzer werden Unregelmäßigkeiten bereits mittels einer

Sichtprüfung ermitteln können. Durch die Auswertung der vom Auditing-System gesammelten Informationen ist auch eine automatisierte Gefahreneinschätzung möglich, die mittels bayesscher Netzwerke in einer Simulation erprobt wurde.

1.4 Aufbau der Arbeit

Im aktuellen Kapitel wird zunächst die Arbeit im Rahmen einer Betrachtung des wissenschaftlich-technischen Kontextes in Bezug zu ähnlichen Arbeiten gesetzt. Im Kapitel 2 werden sodann notwendige Grundlagen und Begriffe eingeführt, wobei das besondere Augenmerk auf der dem Grid zugrundeliegenden PKI und dem Begriff des Auditing liegt. Eine Übersicht der in modernen Grid-Infrastrukturen anzutreffenden Middlewares und Komponenten rundet das Kapitel ab.

Kapitel 3 analysiert die aktuelle Sicherheits- und Bedrohungslage in Grid-Infrastrukturen exemplarisch anhand der nationalen Grid-Initiative *D-Grid*. Die aktuell bestehenden Möglichkeiten zur Detektion und Verhinderung von Credential-Missbrauch werden ebenso beleuchtet wie architekturelle Gegebenheiten. Aus den in diesem Kapitel beleuchteten Problemstellungen wird der im folgenden Kapitel 4 vorgestellte Lösungsansatz entwickelt. Die beispielhafte Implementierung des vorgestellten Lösungsansatzes im Rahmen einer Auditing-Infrastruktur für die deutsche Grid-Initiative ist Inhalt des Kapitels 5.

Mögliche Methoden zur automatisierten Erkennung von Missbrauch bei Grid-Proxy-Credentials werden in Kapitel 6 vorgestellt und evaluiert. Hier wird ebenfalls detailliert auf eine Grid-Simulation eingegangen, die als Machbarkeitsstudie für automatisierte Mißbrauchserkennung diente.

Das Kapitel 7 widmet sich der kritischen Bewertung des zuvor vorgestellten Systems. Der Leser findet in diesem Kapitel eine Analyse der durch die Auditing-Infrastruktur erkennbaren Bedrohungsszenarien und gelöster Problemstellungen. Eine Betrachtung der Leistungsparameter findet ebenfalls in diesem Kapitel statt.

Die gewonnenen Erkenntnisse werden in Kapitel 8 subsummiert und es werden mögliche Erweiterungen und zukünftige Anknüpfungspunkte für das in dieser Dissertation vorgestellte Konzept diskutiert.

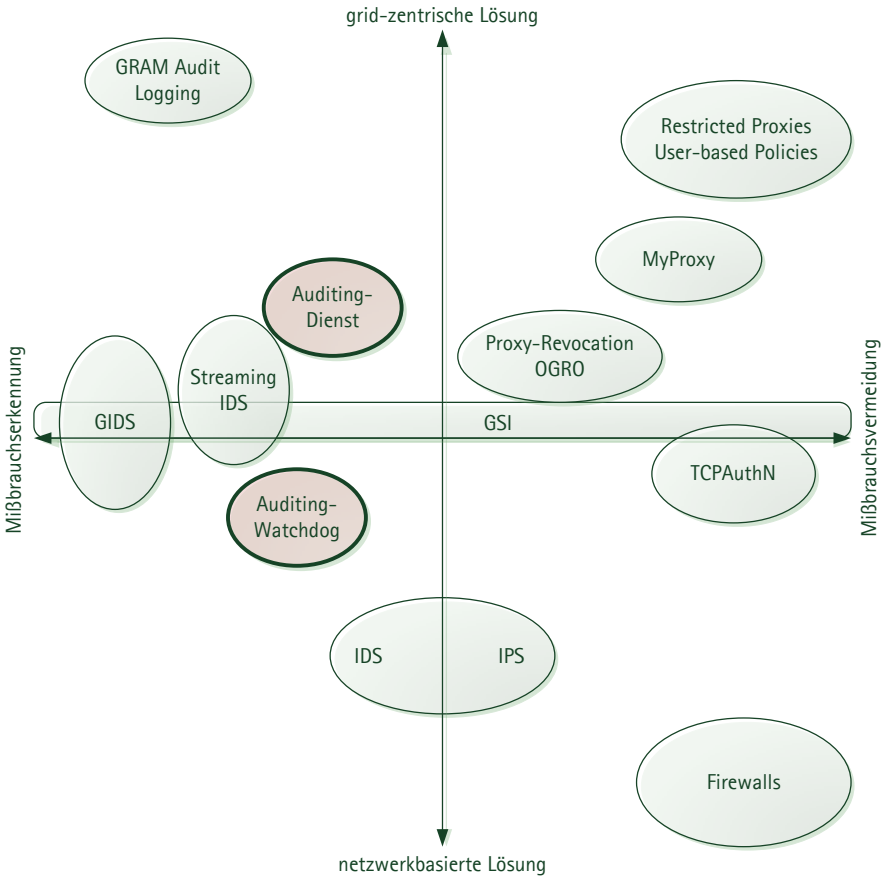


Abbildung 1.1: Ökosystem der Grid- und Netzwerksicherheitskomponenten

1.5 Verwandte Arbeiten

Die vorliegende Arbeit führt eine Komponente ein, die zuvor in verteilten Umgebungen, speziell in Grid-Umgebungen, nicht vorhanden war. Das vorgestellte Konzept ist eine Neuentwicklung, die eine Lücke im Sicherheitskonzept aktueller Grid-Middlewares schließt. Dennoch existieren verschiedene Konzepte und Projekte, die Anknüpfungspunkte an das Auditing-Projekt bieten. Im folgenden Abschnitt soll eine Einordnung des Beitrags dieser Arbeit zum digitalen Ökosystem im Spannungsfeld von Grid-Computing und Netzwerksicherheit vorgenommen werden.

Das Auditing-System fügt sich harmonisch in die aktuelle Grid-Landschaft ein. Es wurde bereits in der Entwurfsphase so konzipiert, daß es mit bestehenden Komponenten vollständig kompatibel ist und nicht durch unverträgliche Änderungen Probleme hervorrufen kann.

Abbildung 1.1 stellt schematisch einige Produkte und Entwicklungen dar, die

mit dem Auditing verwandte Aufgaben im Bereich der Netzwerksicherheit erfüllen. Da der Aufgabenbereich jedes dieser Produkte sich jedoch deutlich von dem des Auditing-Systems unterscheidet, entsteht keine Überlappung mit bestehenden Projekten. Vielmehr wird ein natürlicher, vorhandener Bedarf gedeckt und somit die Sicherheit des Gesamtsystems erhöht. Zusätzlich kann das Auditing in Kooperation mit anderen, bereits vorhandenen oder geplanten Systemen treten, um das Grid noch sicherer zu machen.

Einige Systeme, die Berührungspunkte mit der Auditing-Infrastruktur haben, sollen im Folgenden kurz vorgestellt und ihre Beziehung zum Proxy-Auditing klar abgegrenzt werden.

1.5.1 MyProxy

Bei der Software MyProxy [NTW01], die seit Version 4 fester Bestandteil des Globus-Toolkit ist, handelt es sich um eine Client-Server-Lösung zur Verwaltung von X.509-Zertifikaten und -Schlüsseln. Dabei dient MyProxy sowohl als zentraler Ablageort für Proxy-Credentials (je nach Anwendungsart auch für End-Entity-Credentials) als auch als Online-CA. MyProxy unterstützt neben der Ablage und Ausgabe von Credentials auch die Delegation an vom Nutzer zu bestimmende Entitäten und die Erneuerung abgelaufener Credentials.

Um die Sicherheit der abgelegten Zugangsdaten zu garantieren, wird jeder auf einem MyProxy-Server abgelegte Schlüssel mit einem zuvor vom Nutzer definierten (und bei jeder Verwendung, etwa beim Download eines abgeleiteten Proxy anzugebenden) Passwort verschlüsselt. Somit wird auch den Anforderungen der in der EUGridPMA und TAGPMA organisierten Grid-CAs Rechnung getragen, die eine unverschlüsselte Speicherung von privaten Schlüsseln für längerfristig gültige Zertifikate verbieten.

Durch die Nutzung von MyProxy zur kontrollierten Erstellung von Proxy-Credentials wird dem Nutzer die Bürde abgenommen, sein Nutzerzertifikat und dessen privaten Schlüssel selbst aufbewahren zu müssen – er gibt jedoch auch die Kontrolle über diese wichtigen Authentifizierungswerkzeuge ab. Die Autoren von MyProxy vertreten jedoch die Meinung, dass privater Schlüssel und Nutzerzertifikat auf einem korrekt eingerichteten und abgesicherten MyProxy-Server sicherer seien als auf einem Nutzersystem – somit wird die Gesamtsicherheit im Grid durch MyProxy erhöht.

Ein wichtiges weiteres Leistungsmerkmal der Software ist die Erneuerung abgelaufener Proxy-Credentials. Mittels dieser Funktion können Grid-Ressourcen, die über ein gültiges Credential verfügen, selbsttätig ein neues Credential vom MyProxy-Server beziehen – vorausgesetzt, sie weisen (per gegenseitiger Authentifizierung, siehe Abs. 2.1.3) den Besitz des ablaufenden Credentials nach und stehen in einer vom Nutzer definierten Positivliste berechtigter Ressourcen.

Die Gefahren, die von einem Zertifikatsdiebstahl ausgehen, werden durch MyProxy nicht gemindert. Im Gegenteil – durch geschickte Nutzung des mißbräuchlich erworbenen Credentials sowie der in MyProxy integrierten Funktion zur Zertifikatserneuerung kann ein Angreifer das Zeitfenster deutlich vergrößern, in dem er das Grid mißbräuchlich nutzen kann. Dennoch erfüllt MyProxy einen wichtigen Zweck und wird durch das Auditing sinnvoll ergänzt. Es wurde in der Illustration 1.1 als Komponente zur Missbrauchsverhinderung eingestuft, da ein sicher konfigurierer MyProxy-Server den Missbrauch von End-Entity- und Proxy-Credentials gegenüber einem frei zugänglichen UI-Server deutlich erschwert.

1.5.2 Tracing von Proxy-Delegationen

Die Nutzung von X.509-Erweiterung zur Verfolgung von Zertifikats-Ketten im Grid wurde von der IGTF in einem als experimentell eingestuften Projekt erprobt. Das Dokument „OID for Proxy Delegation Tracing“ [EUG08] definiert eine eindeutige Identifikationskennung (siehe 5.4.1 für detaillierte Informationen zu OIDs), die verwendet wird, um den Weg von Delegationsketten im Grid zu verfolgen. Im Unterschied zu der in der vorliegenden Dissertation vorgestellten Lösung wurde dieses „Tracing“ jedoch lediglich als Werkzeug für die Entwickler von Grid-Middlewares konzipiert, um Fehler im Delegationsprozeß ermitteln und beheben zu können. In produktiven Grid-Anwendungen wird das Proxy-Tracing, wie es von der IGTF definiert wurde, keine Anwendung finden.

1.5.3 Intrusion-Detection-Systeme

Intrusion-Detection-Systeme (IDS) haben die Aufgabe, Angriffe gegen ein Computernetz zu erkennen und – in der Erweiterung als Intrusion Prevention System (IPS) – aktiv abzuwehren. Sie bedienen sich dazu üblicherweise ähnlicher Mittel wie Virenscanner: Eine Datenbank bekannter Angriffssignaturen wird mit dem Da-

tenverkehr im vom IDS überwachten Netz verglichen und Übereinstimmungen lösen einen Alarm aus. Auch heuristische Methoden werden eingesetzt, um nicht durch Signaturen abgedeckte Angriffe erkennen zu können.

Hostbasierte IDS (HIDS) überwachen nur denjenigen Host, auf dem sie gestartet wurden – etwa, indem sie Datenverkehr der Netzwerkschnittstelle mitschneiden oder Logdateien auf Angriffsmuster analysieren. Aktuelle netzwerkbasierte IDS (NIDS) befinden sich hingegen oft am Übergang vom überwachten Netz zum Internet und können so den gesamten ein- und ausgehenden Datenverkehr kontrollieren. Verschlüsselte Daten können in der Regel nicht analysiert werden. Hybriden aus beiden IDS-Typen sind ebenfalls möglich.

Das Auditing stellt eine sinnvolle Ergänzung zu einem IDS dar, da es zwei Fälle abdeckt, die von diesem nicht adressiert werden: Angriffe auf das Grid von innen und unerkannte Angriffe auf das Netzwerk.

Zum Einen sind Angreifer, die bereits über einen legitimen Zugang zum Grid verfügen, diesen aber nicht regelgerecht nutzen, nicht durch eine netzwerkbasierte Angriffserkennung zu entlarven; missbräuchliche Grid-Nutzung kann von einem IDS nicht als solche wahrgenommen werden.

Zum Anderen kann ein IDS nur solche Angriffe detektieren, die auf ein heuristisches Muster oder eine bekannte Signatur passen – und diese Einschränkung stellt eine große Schwäche dar. Moderne Toolkits zum Angriff oder „Penetration Testing“ verteilter Infrastrukturen wie etwa Metasploit[1] nutzen diese Schwäche gezielt aus, indem sie mittels komplexer Verschleierungstechniken den durch einen Angriff erzeugten Netzwerktraffic so maskieren, dass weder signatur- noch heuristikbasierte IDS diesen als Angriff erkennen können. Schlüpft ein Angreifer so durch die Maschen der netzwerkbasierten Intrusion Detection und verschafft sich Zugriff auf Grid-Proxy-Zertifikate, bietet das Proxy-Auditing eine zusätzliche Möglichkeit, Angriffe zu erkennen und den angegriffenen Nutzer zu benachrichtigen.

Streaming IDS

Für die speziellen Anforderungen von Grid-Infrastrukturen, besonders im Hinblick auf das hohe zu analysierende Verkehrsaufkommen, wurde in einer Veröffentlichung [SSH+09] ein „Streaming Intrusion Detection System“ konzipiert, dessen Komponenten auf die Erkennung von Angriffen über Site-Grenzen hinweg ausgelegt sind.

[1] `http://www.metasploit.com/learn-more/what-is-it/`

Das Streaming IDS behandelt alle untersuchten Daten als kontinuierlichen Datenstrom; die aggregierten Datenströme aller Grid-Sites werden als Eingabedaten für die Erkennungsregeln verwendet. Die Aggregation der Rohdaten erleichtert – so die Autoren – die Erkennung von Angriffen, die sich über mehrere Sites erstrecken (wie in Abschnitt 3.4.1 geschildert).

GIDS

Das D-Grid-Projekt „GIDS - ein Grid-basiertes, föderiertes Intrusion Detection System zur Sicherung der D-Grid-Infrastruktur“ [HgE+10] verfolgt die Zielsetzung, einen Verbund von Intrusion-Detection-Systemen zu schaffen und zu verwalten, der jede D-Grid-Site abdeckt. Dazu sollen die HIDS/NIDS, die bereits von jedem D-Grid-Ressourcenprovider betrieben werden, ihre Funde zusätzlich zu den vorhandenen Meldewegen noch an einen „GIDS-Agenten“ weiterleiten, der sie dann an einen zentralen Nachrichten-Bus weiterleitet. Über spezielle Filter können Ressourcenbetreiber festlegen, welche Meldungen ihres lokalen IDS vom Agenten an den GIDS-Bus weitergeleitet werden und ob diese anonymisiert/pseudonymisiert werden müssen.

Die Architektur und der Leistungsumfang des GIDS spiegeln deutlich die heterogenen Anforderungen des D-Grid wider: Der föderale Gedanke der nationalen Grid-Infrastruktur wird aufgegriffen und genutzt, während die Ressourcenbetreiber weiterhin frei entscheiden können, welches lokale IDS zum Einsatz kommt und welche Warnmeldungen an die zentrale Instanz gemeldet werden sollen. Mittels eines föderierten IDS können zudem Angriffe erkannt werden, die sich über mehrere angeschlossene Sites erstrecken. Eine Anbindung des GIDS an das VO-Management des D-Grid ist ebenfalls beabsichtigt.

Nicht Teil des GIDS-Konzeptes ist hingegen die Erkennung grid-spezifischer Angriffe; die Detektionskomponenten der ressourcen-lokalen IDS werden nicht modifiziert. Daher ist GIDS als Ergänzung und Erweiterung eines bereits vorhandenen IDS zu sehen, nicht als Konkurrenzprodukt für das Auditing. Zur Verknüpfung beider Systeme ist eine Schnittstelle denkbar, damit Fälle von Proxy-Missbrauch an das GIDS gemeldet und von dort zu den betroffenen Nutzern und Ressourcenbetreibern weitergeleitet werden können.

1.5.4 Firewalls

Firewalls sind Netzwerkgeräte, die am Übergang zwischen Netzen mit unterschiedlichen Sicherheitsbedürfnissen oder -stufen plaziert werden, um beide Netze gegen unerwünschte Zugriffe aus dem jeweils anderen Netz abzugrenzen. Als Basis für diese Abgrenzung dient die Definition von Regeln anhand der Quell- und Zieladressen und -ports von IP-Paketen.

Im Unterschied zu einem IDS und auch dem Auditing-Konzept besteht die Funktion einer Firewall nicht darin, Angriffe zu erkennen; sie soll lediglich aufgrund des ihr zur Verfügung stehenden Regelwerks unerwünschte Netzwerkverbindungen ohne Ansehen ihrer Legitimität verhindern. Zudem sind Angriffe auf eine Grid-Infrastruktur in der Regel so beschaffen, dass sie genau jene Netzwerkdienste Missbrauchen, die der Außenwelt zur Verfügung gestellt werden – eine Firewall bietet hier nur selten einen Schutz. Angriffe, die von böswilligen Grid-Nutzern innerhalb einer Grid-Site durchgeführt werden, werden darüber hinaus überhaupt nicht erfaßt.

Die Betriebskonzepte vieler Grid-Infrastrukturen – u.A. des D-Grid (siehe [VG07]) – sehen eine Firewall als wichtigen Schutzmechanismus gegen Angriffe vor. Tatsächlich schließen Firewalls wirksam eine Vielzahl von Angriffsvektoren. Dennoch ist ein vollständiger Schutz gegen Angriffe von innen und außen auch mit einer leistungsfähigen Firewall nicht möglich und die Überwachung von Grids auf Proxy-Missbrauch durch das Auditing stellt einen zusätzlichen Sicherheitsgewinn dar.

TCP AuthN

Der Nutzwert von Firewalls als Absicherung einer Grid-Site gegen Angriffe von außen ist stark abhängig von der Möglichkeit, unerwünschten Datenverkehr anhand seiner Quell- und Zieldresse auszuschließen. Im Grid ist dieses Vorgehen jedoch aufgrund technischer Gegebenheiten, insbesondere für den hochperformanten Datentransfer zwischen Sites, nur eingeschränkt möglich. Aktuelle Betriebskonzepte sehen die unkonditionale Freigabe großer Portbereiche vor, die von Angreifern dann auch für ihre Zwecke genutzt werden können. Das in [WKPG09] vorgestellte Konzept modifiziert den TCP-Verbindungsaufbau derart, dass Authentifizierungsinformationen des Nutzers bereits während des Aufbaus einer Verbindung an die Firewall übertragen werden, indem eine signierte Referenz auf ein X.509-Zertifikat versendet wird. Die Firewall ist somit bereits beim Aufbau einer TCP-Verbindung in der Lage, den Nut-

zer zu identifizieren und kann den Aufbau entsprechend dessen Autorisierungsstufe gestatten oder verhindern.

1.5.5 OGRO

Das Projekt „Open GRid Ocsp“ (OGRO), vorgestellt von Luna et al. in [LMM05], [LMM07] und [Lun08], adressiert das Problem fehlender Zertifikats-Revocation in Grid-Infrastrukturen. Wie in 2.2.4 erläutert, können Proxyzertifikate in Grid-Infrastrukturen nicht zurückgezogen werden, da keine zentrale vertrauenswürdige Instanz existiert, die eine entsprechende Liste verwalten könnte. Das OGRO-Konzept implementiert eine OSCP-Infrastruktur in einem Globus-basierten Grid, indem zum Einen ein OCSP-Dienst entwickelt wurde, der von Grid-Nutzern mit Informationen über zurückgezogene Proxyzertifikate versorgt wird; zum Anderen haben die Entwickler die GSI-Bibliotheken modifiziert, um diesen Dienst während eines Authentifizierungsvorgangs abzufragen.

Mittels einer funktionierenden Revocation von Proxyzertifikaten wäre es möglich, die Angriffserkennung des Auditing-Systems mit einer wirkungsvollen Möglichkeit zur Missbrauchsabwehr auszustatten. Leider ist das OGRO-Projekt seit geraumer Zeit inaktiv, was eine Anbindung an das Auditing-System unmöglich gemacht hat.

1.5.6 User-based Policies

In einer 2009 veröffentlichten Dissertation [Pig08] wird ein Konzept zur Einschränkung der Nutzerprivilegien in Grid-Infrastrukturen vorgestellt. Dieses Konzept verwendet XACML-Policies, die als X.509-Erweiterungen in Proxyzertifikate eingebettet werden. Nutzer können mithilfe dieser Policies festlegen, welche Aktionen mithilfe eines Proxyzertifikats erlaubt sind. So kann die Abgabe und Manipulation von Grid-Jobs auf die in der entsprechenden Policy genannten Job-Identifikatoren eingeschränkt werden; ebenso werden Datenzugriffe per GridFTP oder dCache eingeschränkt. Werden derart eingeschränkte Proxyzertifikate von Unbefugten Missbraucht, ist das Schadenspotential deutlich geringer. Die Daten- und Job-Policies werden direkt auf den betroffenen Ressourcen ausgewertet; zu diesem Zweck wurden die dortigen Autorisierungsabläufe modifiziert.

Das Konzept der Einschränkung delegierter Rechte stellt einen Sicherheitsgewinn für Grid-Infrastrukturen dar, räumt aber das inhärente Missbrauchspotential de-

legierter Rechteausübung nicht vollständig aus. Angreifer können auch mit einem eingeschränkten Proxy-Credential diejenigen Daten und Jobs manipulieren und ggf. zerstören, für die das Credential ursprünglich ausgestellt wurde. Zusätzlich zur Einschränkung der an eine Delegation eingeräumten Rechte ist also eine Überwachung und Benachrichtigung des Betroffenen bei Missbrauch in jedem Fall notwendig.

1.5.7 Logging und Accounting in Grid-Middlewares

Der Begriff des „Auditing“ wird – auch im Grid-Kontext – häufig verwendet, um die Erfassung von Nutzungsdaten zu bezeichnen, die als Grundlage für eine spätere Auflistung und Abrechnung der genutzten Ressourcen dienen können. Diese Erfassung geschieht häufig mit Rückgriff auf Komponenten zum Logging, also zur Erfassung administrativ relevanter Informationen während des Grid-Betriebs. Die Begriffe des Logging und Accounting sollen nun im Kontext aktueller Grid-Middlewares kurz eingeordnet werden.

Logging ist das Erzeugen und Speichern eines Protokolls zu Diagnose- und Nachweiszwecken. In ihrer Veröffentlichung zu semantischem Logging im Grid [BB05] unterscheiden Baker und Boakes fünf verschiedene Logtypen, und zwar System, Application, Communication, Specialised Instrumentation und Environment Monitor Logs. Logdateien stellen keine aggregierten, sondern Rohdaten zur Verfügung, bilden also eine der Grundlagen für Auditingvorgänge.

Berührungspunkte mit dem Auditing haben auch die Erfassung (*Accounting*) und Abrechnung (*Billing*) von Nutzungsdaten. Das Accounting in Grid-Infrastrukturen hat zum Ziel, die Nutzung von Ressourcen (wie etwa CPU-Zeit, Netzwerkverkehr und Speicherplatz) dem jeweiligen Nutzer zuzuordnen – dies geschieht anhand seiner Authentifizierungsinformationen. Besteht (etwa wegen der unerlaubten Duplizierung von Grid-Credentials) Zweifel an der Authentizität dieser Informationen, so werden Auditing und Billing fehlerhaft und somit nicht mehr verläßlich.

Die Middlewares Globus und gLite nutzen für diese Anforderungen jeweils eigene Komponenten, die als „GRAM Audit Logging“ (Globus Toolkit) und „Logging & Bookkeeping Service“ (gLite) bezeichnet werden.

GRAM audit logging

Unter der Bezeichnung „GRAM Audit Logging“ [The07] enthält der Globus Resource Allocation Manager (GRAM) eine Komponente zur Erfassung der Vitaldaten von Grid-Jobs. Diese Komponente ist jedoch nicht dafür konzipiert, nutzerzentrische Informationen zu aggregieren, sondern dient als vorbereitende und unterstützende Komponente für das Accounting. Der Audit-Logging-Dienst von Globus speichert sämtliche Informationen eines Jobs, die für die korrekte Verbuchung notwendig sind, also neben einer Nutzeridentifikation vor allem Informationen über die Laufzeit und den Ressourcenbedarf. Informationen über die vom Nutzer delegierten Credentials werden durch das „Audit Logging“ nicht verarbeitet oder gespeichert; die gespeicherten Einträge sind gegen nachträgliche Manipulation nicht geschützt.

Das GRAM Audit Logging wurde nicht für die Abfrage durch Endnutzer konzipiert; es steht somit keine Schnittstelle zur Verfügung, um von Nutzerseite die vollständigen vom Audit Logging gespeicherten Informationen abzufragen. Als Werkzeug zur Erhöhung der Transparenz bei der Grid-Nutzung scheidet es daher aus. In Abbildung 1.1 wird es dennoch als mögliche Lösung zur Missbrauchserkennung aufgeführt, da dieser Einsatzzweck theoretisch denkbar wäre.

gLite logging & bookkeeping

Die auf Globus 2 basierende gLite-Infrastruktur hat eigene Verfahren zum Accounting von Jobs und Datennutzung. Dabei kommt ein mehrteiliges System zum Einsatz, das als „Logging und Bookkeeping“ bezeichnet wird und auf der gLite-Homepage[2] näher beschrieben ist. Das „L&B“ agiert ereignisgesteuert und sammelt sog. „Events“ für jeden Job lokal an den Stellen, an denen diese Ereignisse generiert werden; insbesondere also am Workload Management System (WMS) und dem Computing Element (CE).

Wie auch das GRAM Audit Logging ist der L&B nicht zur Überprüfung auf mißbräuchliche Nutzung geeignet.

[2] http://glite.web.cern.ch/glite/lb/

Kapitel 2

Grundlagen

Die technischen Grundlagen für die in dieser Dissertation vorgestellten Konzepte und Entwicklungen werden im folgenden Kapitel eingeführt. Dazu zählen insbesondere die Grundlagen asymmetrischer Kryptographie, eine Abgrenzung der Begriffe Authentisierung, Authentifizierung, Autorisierung und Grundbegriffe des Grid-Computing sowie ein Überblick über aktuelle Grid-Initiativen und -Forschungsprojekte im europäischen Kontext.

Des Weiteren wird der Begriff des „Auditing“ eingeführt und seine verschiedenen Bedeutungsebenen in unterschiedlichen Bereichen diskutiert.

2.1 Kryptographie und Authentifizierung

2.1.1 Verschlüsselung und Signatur

Kryptographie (griechisch: kryptós, „verborgen“ und gráphein, „schreiben“) dient der Verschlüsselung von Nachrichten mittels eines Algorithmus und eines oder mehrerer Schlüssel. Ein sicherer kryptographischer Algorithmus schützt Informationen, indem er Vertraulichkeit und Integrität sowie – mithilfe von Verfahren zur digitalen Signatur – Authentizität und Verbindlichkeit gewährleistet. Während die symmetrische Kryptographie auf der Verschlüsselung mit einem zuvor über einen sicheren Kanal oder mittels eines Schlüsselaustauschverfahrens allen Kommunikationspartnern mitgeteilten Schlüssels beruht, wird in der asymmetrischen Kryptographie ein Schlüsselpaar eingesetzt, das aus einem öffentlichen und einem geheimen Schlüssel besteht.

Symmetrische Kryptographie

Symmetrische Kryptographieverfahren, etwa der *Data Encryption Standard* (DES) und der *Advanced Encryption Standard* (AES) [Ver01] verwenden denselben Schlüssel K zur Ver- und Entschlüsselung des Klartextes M zu einem Ciphertext C. Es existiert also eine Verschlüsselungsfunktion

$$E_K(M) = C$$

und eine Entschlüsselungsfunktion

$$D_K(C) = M$$

mit der Eigenschaft

$$D_K(E_K(M)) = M.$$

Zentraler Nachteil herkömmlicher symmetrischer Verschlüsselungsverfahren ist die Notwendigkeit, den zum Ver- und Entschlüsseln notwendigen Schlüssel vor der gesicherten Kommunikation allen Teilnehmern zu übermitteln. Zu diesem Zweck existieren Verfahren zum Austausch eines Schlüssels über einen ungesicherten Kanal, etwa das Diffie-Hellman-Protokoll [DH76b].

Ein großer Vorteil symmetrischer Kryptographiealgorithmen gegenüber asymmetrischer Kryptographie ist die deutlich höhere Geschwindigkeit bei Ver- und Entschlüsselung, weswegen in modernen Kryptosystemen (etwa TLS/SSL) häufig aus Leistungsgründen ein hybrider Ansatz verwendet wird.

Asymmetrische Kryptographie

Das Konzept der asymmetrischen oder Public-Key-Kryptographie wurde 1976 von Whitfield Diffie und Martin Hellman vorgestellt [DH76b] [DH76a] und in vielen Protokollen und Algorithmen implementiert. Grundsätzlich eignet sich asymmetrische Kryptographie sowohl zum Signieren von Nachrichten als auch zu deren Verschlüsselung; allerdings sind nur wenige Algorithmen für beide Aufgaben geeignet, darunter RSA [RSA78] [RSA77] und ElGamal [ElG85].

Bei symmetrischen Verschlüsselungsverfahren wird zum Ver- und Entschlüsseln einer Nachricht jeweils ein unterschiedlicher Schlüssel verwendet. Der zum Verschlüsseln notwendige öffentliche Schlüssel K_1 wird jedem Kommunikationspartner mitgeteilt und der zum Entschlüsseln notwendige private Schlüssel K_2 verbleibt in der Obhut des Inhabers. Somit ergeben sich für das Kryptosystem folgende Eigenschaften:

$$E_{K1}(M) = C$$
$$D_{K2}(C) = M$$
$$D_{K2}(E_{K1}(M)) = M$$

Kryptographisch sicher wird ein asymmetrischer Verschlüsselungsalgorithmus ausschließlich durch die Verwendung eines geeigneten Schlüsselpaars und der Geheimhaltung der privaten Schlüssel. Details zum Algorithmus helfen einem Angreifer ebenso wenig bei der Entschlüsselung einer abgefangenen Nachricht wie der Besitz eines oder mehrerer öffentlicher Schlüssel.

Im heute gebräuchlichsten Verfahren zur asymmetrischen Verschlüsselung, RSA, wird das Schlüsselpaar aus Primzahlen erzeugt und damit der Umstand ausgenutzt, dass die Faktorisierung großer Zahlen auch bei Einsatz hoher Rechenleistung sehr aufwendig ist. Es kann somit davon ausgegangen werden, dass ein außenstehender Angreifer den privaten Schlüssel nicht aus dem öffentlichen errechnen und somit auch die Nachricht nicht entschlüsseln kann.

Hybride Ansätze

Um die Vorteile symmetrischer Kryptographie – Geschwindigkeit und Robustheit – mit denen der asymmetrischen Kryptosysteme – insbesondere dem Verzicht auf ein Schlüsselaustauschprotokoll oder out-of-band-Kommunikation – zu verbinden, werden in der Praxis vielfach hybride Ansätze verfolgt. Typischerweise wird in diesen Ansätzen zunächst vom Initiator einer Kommunikation ein zufälliger Session-Schlüssel erzeugt. Dieser wird dann mittels asymmetrischer Kryptographie verschlüsselt (ggf. noch signiert) und dem Kommunikationspartner übermittelt. Dieser kann nun die weitere Kommunikation unter Verwendung des Session-Schlüssels mithilfe eines symmetrischen Verfahrens verschlüsseln. TLS und SSL nutzen einen solchen hybriden Ansatz.

Digitale Signatur

Während Verschlüsselung zwar die Integrität und Vertraulichkeit von Nachrichten sicherstellt, kann ein Kommunikationspartner nicht feststellen, ob die Nachricht authentisch ist, also tatsächlich vom angenommenen Urheber stammt. Da zum Verschlüsseln der öffentliche Schlüssel verwendet wurde, könnte auch jeder beliebige Dritte eine Nachricht mit diesem Schlüssel chiffrieren. Aus dieser mangelnden Authentizität folgt ebenfalls, dass die Urheberschaft einer Nachricht vom angenommenen Urheber abgestritten werden kann und somit die Verbindlichkeit („Non-Repudiation“) der Nachricht nicht gewährleistet ist.

Abhilfe schaffen hier digitale Signaturen, die an einer Nachricht angebracht werden. Zu diesem Zweck bildet der Versender der Nachricht zunächst mittels eines geeigneten, sicheren Ein-Wege-Verfahrens (etwa SHA [EJ01]) eine Prüfsumme über die gesamte Nachricht. Diese Prüfsumme verschlüsselt er sodann mit seinem privaten Schlüssel K_{priv} und teilt sie zusammen mit der Nachricht und dem verwendeten Prüfsummenverfahren dem Empfänger mit. Dieser, der über den öffentlichen Schlüssel des Versenders verfügt, kann nun zunächst mit seinem eigenen privaten Schlüssel die Nachricht und dann mithilfe des öffentlichen Schlüssels des Versenders die verschlüsselte Prüfsumme entschlüsseln. Vergleicht er die so erhaltene Klartext-Prüfsumme mit der Prüfsumme, die er mithilfe des zuvor vom Versender mitgeteilten Verfahrens über die entschlüsselte Nachricht gebildet hat, so kann der Empfänger feststellen, ob die Nachricht im Transit modifiziert wurde und ob sie vom Versender stammt.

Nur dieser ist nämlich in der Lage, eine Nachricht mit seinem privaten Schlüssel zu chiffrieren.

Somit gewährleisten Verfahren zur digitalen Signatur die beiden wichtigen Eigenschaften der Verbindlichkeit und Authentizität, die im Kontext dieser Arbeit von besonderem Gewicht sind.

2.1.2 Zertifikate und PKI

Zwar kann durch Anwendung digitaler Signaturen festgestellt werden, ob eine Nachricht vom angegebenen Absender stammt – es ist jedoch nicht möglich, die Identität dieses Absenders festzustellen. Diese Lücke in der sicheren Kommunikation wird in modernen Kryptosystemen häufig durch den Einsatz von digitalen Zertifikaten geschlossen. Ein solches Zertifikat enthält die Identitätsinformationen einer Entität (also typischerweise eines Nutzers oder eines Rechnersystems) sowie deren öffentlichen kryptographischen Schlüssel. Es wird von einer universell als vertrauenswürdig anerkannten Stelle (CA, s.u.) mit deren privatem Schlüssel signiert. Diese Stelle zertifiziert damit den Zusammenhang zwischen Identitätsinformationen und öffentlichem Schlüssel.

Möchte nun ein Kommunikationspartner die Urheberschaft einer Nachricht prüfen, kann er das anhand des digitalen Zertifikats, das den zur Signatur der Nachricht passenden Schlüssel enthält, tun. Stimmen die Schlüssel überein und vertraut der Kommunikationspartner der Zertifizierungsstelle, so kann er mit ausreichender Sicherheit annehmen, die Identität des Urhebers der Nachricht zu kennen.

Digitale Zertifikate inner- und außerhalb des Grid werden gemäß der ITU-T-Empfehlung X.509 [ITU05] erstellt und verfügen über standardisierte Attribute. So besitzt ein X.509-Zertifikat stets eine Laufzeit, die in der Regel für Nutzer- und Hostzertifikate ein Jahr beträgt, aber auch deutlich länger oder kürzer sein kann. Ein Zertifikat kann nicht länger gültig sein als das Zertifikat des Ausstellers; wird dieses ungültig, verliert auch jedes nachrangige Zertifikat seine Gültigkeit.

Jedes Zertifikat bekommt eine CA-weit eindeutige Seriennummer, anhand der es referenziert werden kann. Diese Seriennummer wird unter Anderem genutzt, um zurückgezogene Zertifikate zu identifizieren.

```
1 Certificate:
    Data:
3       Version: 3 (0x2)
```

```
        Serial Number: 13344 (0x3420)
5       Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=DE, O=GermanGrid, CN=GridKa-CA
7       Validity
            Not Before: Jan 17 15:03:07 2011 GMT
9           Not After : Feb 16 15:03:07 2012 GMT
        Subject: C=DE, O=GermanGrid, OU=UniHannover, CN=Christopher
            Kunz
11      Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
13          RSA Public Key: (2048 bit)
                Modulus (2048 bit):
15                  00:ab:c2:d0:6f:e3:ea:ec:74:3e:49:9d:df:cf:77:
    [..]
17                  38:b7
                Exponent: 65537 (0x10001)
19      X509v3 extensions:
            X509v3 Basic Constraints: critical
21              CA:FALSE
            X509v3 Key Usage: critical
23              Digital Signature, Non Repudiation, Key Encipherment,
                    Data Encipherment
            X509v3 Subject Key Identifier:
25              80:13:F3:88:BC:0D:4F:A2:77:67:D1:05:DC:02:B5:15:2C:23:6
                    C:46
            X509v3 Authority Key Identifier:
27              keyid:C6:75:C9:28:AC:D1:0B:FC:3C:FF:B9:B5:1E:D3:5F:3B
                    :80:62:12:34
                DirName:/C=DE/O=GermanGrid/CN=GridKa-CA
29              serial:00

31          X509v3 Subject Alternative Name:
                email:kunz@dcsec.uni-hannover.de
33          X509v3 Issuer Alternative Name:
                email:gridka-ca@iwr.fzk.de
35          X509v3 CRL Distribution Points:
                URI:http://grid.fzk.de/ca/gridka-crl.der
37
            X509v3 Certificate Policies:
39              Policy: 1.3.6.1.4.1.2614.5548.1.1.1.5
```

```
41              Netscape Cert Type:
                    SSL Client, S/MIME
43              Netscape Comment:
                    Certificate issued under CP/CPS v. 1.5 at http://grid.
                        fzk.de/ca
45              Netscape Base Url:
                    http://grid.fzk.de/ca
47              Netscape CA Policy Url:
                    http://grid.fzk.de/ca/gridka-cps.pdf
49              Netscape Revocation Url:
                    http://grid.fzk.de/ca/gridka-crl.der
51      Signature Algorithm: sha1WithRSAEncryption
            0a:51:ce:00:0c:c3:a1:11:39:1f:80:e1:f3:d0:1c:c7:7f:bb:
53          [..]
            13:40:83:ae
```

Listing 2.1: Ein X.509-Zertifikat (gekürzt)

Die Inhaber- und Ausstellerinformationen in einem X.509-Zertifikat werden gemäß des Standards X.500 [ITU08a] als Distinguished Name, kurz DN, hinterlegt. Diese Namen sind im Namensraum der Zertifizierungsstelle „distinguished“, also eindeutig unterscheidbar und werden somit nur einmal verwendet. Ein beispielhafter Distinguished Name wäre der Inhabername des obenstehenden Zertifikats (Zeile 10):

```
Subject: C=DE, O=GermanGrid, OU=UniHannover, CN=Christopher Kunz
```

Hierbei stehen die Werte von C und O für Land (Country) und Organisation, OU für eine Unterorganisation (Organizational Unit) und CN für den „Common Name“, also den gebräuchlichen Namen des Zertifikatsinhabers. Dieser ist bei Nutzerzertifikaten üblicherweise der im Ausweis verzeichnete Name des Inhabers, bei Hostzertifikaten jedoch der voll qualifizierte Hostname (FQDN) des betreffenden Rechners. CN-Elemente können in einem Zertifikat mehrfach vorkommen.

Die Namenskonventionen für ein CA-Zertifikat sind im Wesentlichen analog zu denen eines EEC; der DN eines CA-Zertifikats muß jedoch weltweit eindeutig sein (da der Namensraum, in dem sich CAs befinden, global ist).

Zertifikate können neben den standardisierten Attributen auch zusätzliche erweiterte Informationen enthalten; diese werden in Zertifikatserweiterungen (Extensions) gespeichert, die anhand einer OID (siehe Abs. 5.4.1) identifiziert werden. Extensions

können beliebige Informationen, auch Binärdaten, enthalten.

Wird der private Schlüssel zu einem Zertifikat kompromittiert, wurde das Zertifikat irrtümlich ausgestellt oder in betrügerischer Absicht beantragt, so kann die Zertifizierungsstelle es zurückziehen. Dieser Vorgang wird als *Revocation* bezeichnet. Da die CA jedoch die ausgegebenen Zertifikatsdateien nicht physisch zurückfordern kann, hinterlegt sie die Seriennummer und den Fingerprint (also eine textuelle Repräsentation des öffentlichen Schlüssels) in einer speziellen Liste, der „Certificate Revocation List“. Diese Liste wird im Internet veröffentlicht; ihre URL wird in jedem Zertifikat hinterlegt. Im oben stehenden Listing 2.1 ist die CRL-URL folgendermaßen gespeichert (Zeile 49):

```
X509v3 CRL Distribution Points:
URI:http://grid.fzk.de/ca/gridka-crl.der
```

Zertifikatsverarbeitende Stellen müssen stets eine aktuelle Version der CRL für jede CA vorhalten.

Neuere Ansätze zur Überprüfung des Zertifikatsstatus verwenden hingegen das Online Certificate Status Protocol OCSP [MAM+99], das einen sofortigen Online-Abgleich mit der Zertifizierungsstelle durchführt und somit auch vor sehr kurzer Zeit zurückgezogene Zertifikate berücksichtigen kann.

Chain of Trust

Sinn und Zweck digitaler Zertifikate ist es, eine Identität an ein Schlüsselpaar zu binden und somit in verteilten Systemen eine Möglichkeit zur Feststellung zu bieten, ob eine Entität tatsächlich ist, wer sie zu sein behauptet. Dazu ist es notwendig, dass alle Nutzer der CA vertrauen, da diese die Bindung zwischen Identität und Schlüsselpaar mit ihrer digitalen Signatur bestätigt. Um die Gültigkeit eines Zertifikats zu prüfen, wird also das Ausstellerzertifikat benötigt. Dessen öffentlicher Schlüssel wird benutzt, um die Signatur des zu prüfenden Zertifikats zu verifizieren; ist diese gültig, kann die Echtheit des Nutzerzertifikats vermittels des Vertrauensverhältnisses zur CA vorausgesetzt werden.

In der Regel hat diese Vertrauenskette oder „Chain of Trust“ selten mehr als zwei oder drei Elemente: Das CA-Zertifikat wird bisweilen genutzt, um ein Zwischenzertifikat (Signing Certificate) zu signieren; dieses wiederum signiert die Nutzerzertifikate. In Grid-Infrastrukturen werden hingegen noch weitere Elemente angehängt, die

vom Nutzerzertifikat signiert werden. Diese sogenannten „Proxy-Zertifikate“ werden in Abschnitt 2.2.4 detailliert erläutert.

Public Key Infrastructures

Der Begriff Public-Key-Infrastruktur (PKI) bezeichnet ein System, das digitale Zertifikate auf eine sichere Art und Weise vergeben und dabei die Identität der Empfänger bestätigen kann. Kopf der PKI ist die Certificate Authority (CA, Zertifizie-

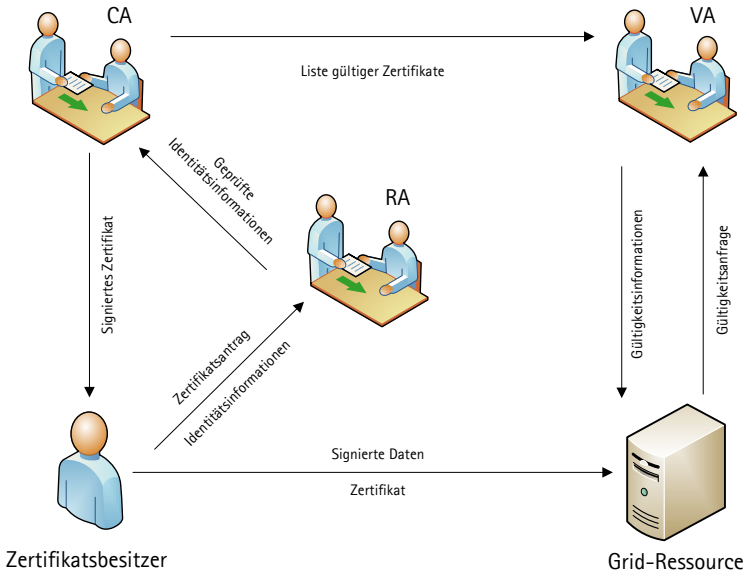


Abbildung 2.1: Übersicht über eine Grid-PKI

rungsstelle). Sie stellt digitale Zertifikate aus, garantiert also die Authentizität und Validität der in ihnen enthaltenen Informationen mit ihrer digitalen Signatur. Diese Signatur wird mit dem zum Zertifikat der CA gehörigen privaten Schlüssel erstellt und kann somit mithilfe des CA-Zertifikats leicht geprüft werden[1].

Somit stellt die CA die Wurzel der gesamten Vertrauenskette in der PKI dar; eine Stellung, die meist nicht nur durch technische, sondern auch umfangreiche organisatorische oder politische Vorgänge erreicht wird. So werden etwa die Zertifizierungsstellen, die in aktuellen Webbrowsern als „vertrauenswürdig“ gekennzeichnet sind, regelmäßigen Sicherheitsaudits unterzogen und müssen zudem die Nachvollziehbarkeit und Validität ihrer Vergabepraxis nachweisen. Die für europäische Grid-

[1]Der private Schlüssel des CA-Zertifikats ist somit ein „Single Point of Failure“ einer PKI – gelangt er in unbefugte Hände oder wird ungültig, müssen alle Zertifikate neu ausgestellt werden.

Infrastrukturen vertrauenswürdigen Zertifizierungsstellen werden von der European Grid Policy Management Authority (EUGridPMA [EUG06]) anhand eines Richtlinienkatalogs festgelegt. Jeder Teilnehmer an einer Grid-Infrastruktur vertraut somit denselben CAs wie seine Kommunikationspartner und hat somit stets eine vertrauenswürdige Stelle, die als „Vertrauensanker“ (engl. Trust Anchor) für die Authentifizierung dient.

Neben der CA enthält eine PKI typischerweise noch eine Registrierungsstelle (Registration Authority, RA), die Zertifikatsanfragen bearbeitet und die Identität des Antragstellers verifiziert. Grid-CAs unter der Ägide der EUGridPMA verlangen typischerweise eine persönliche Identifikation mittels Ausweisdokumenten, um einen Zertifikatsantrag einer natürlichen Person zu genehmigen.

Neben dem natürlichen Ablaufen durch Ende der Gültigkeitsperiode kann ein Zertifikat auch von der Zertifizierungsstelle zurückgezogen (revoked) werden – bei Verlust oder unerlaubter Vervielfältigung des Private Key oder bei anderem Missbrauch ist dieses Vorgehen üblich. Die Verifizierungsstelle (VA, Verification Authority) kann von Teilnehmern einer PKI angerufen werden, um mittels eines geeigneten Protokolls wie etwa OCSP die Gültigkeit eines Zertifikats zu prüfen.

2.1.3 Authentifizierung

Authentifizierung (engl. Authentication, kurz AuthN) ist die Verifizierung einer behaupteten Eigenschaft einer Partei durch eine prüfende Partei - insbesondere der Nachweis, dass die zu prüfende Partei die ist, die sie zu sein vorgibt. Die Authentifizierung der Parteien einer Kommunikation ist eine Grundbedingung für deren Vertraulichkeit. Im Grid-Kontext wird eine Authentifizierung während des Verbindungsaufbaus vor praktisch jeder Kommunikation zwischen Grid-Komponenten und -Nutzern durchgeführt.

In der deutschen Sprache existieren zwei Termini, die im Englischen mit „Authentication“ übersetzt werden. Beide bezeichnen jeweils verschiedene Aktionen während des Authentifizierungsvorgangs zwischen der zu authentifizierenden Partei A und der prüfenden Partei B: Zur *Authentisierung* wird von Partei A eine Information übermittelt, die nur sie wissen kann (z.B. Passwort oder verschlüsselte Zufallszahl, siehe Abschnitt 2.1.3). Partei B überprüft diese Information mittels ihr zur Verfügung stehender Mittel (Prüfung des Passworts mittels einer Datenbank, Entschlüsselung und Überprüfung der verschlüsselten Zufallszahl) - dieser Vorgang wird als *Authen-*

tifizierung bezeichnet.

Da jedoch in der überwiegend englischsprachigen Literatur der Begriff „authentication“ für den Gesamtvorgang der Authentisierung und Authentifizierung steht, wird diese Terminologie im weiteren Verlauf übernommen.

Praktisch jede privilegierte Aktion in einem Mehrbenutzer-Computersystem setzt eine Authentifizierung des jeweiligen Anwenders voraus. Dieser kann sich mittels mehrerer Faktoren authentifizieren:

- Mittels eines Gegenstandes, den er *besitzt*, etwa eine Smart-Card, eine Schlüsseldatei o.ä.
- Über eine Information, die er *weiß*, etwa ein Passwort
- Über etwas, das er *ist*, also durch Prüfung biometrischer Merkmale wie Fingerabdruck oder Retinamuster.

Authentifizierung mittels eines dieser Faktoren wird als Ein-Faktor-Authentifizierung, jene mittels multipler Faktoren folglich als Mehr-Faktor-Authentifizierung bezeichnet. Je mehr Faktoren benötigt werden, desto sicherer kann die Authentizität des Subjekts angenommen werden. So ist bei der Bezahlung von Waren mittels elektronischer Scheckkarte eine Zweifaktor-Authentifizierung (Besitz der Karte und Wissen um die PIN-Nummer) üblich, während Autos und Wohnungen üblicherweise noch immer mittels Einfaktor-Authentifizierung (Besitz des passenden Schlüssels) zugänglich sind.

In Grid-Infrastrukturen ist es unpraktikabel, die Faktoren „Wissen“ und „persönliche Merkmale“ bei jeder Interaktion mit dem Grid neu abzufragen, daher wird hier die Authentizität einer Anfrage über den Besitz eines privaten Schlüssels sichergestellt.

Authentifizierung in Public-Key Infrastrukturen

Die meisten Verfahren zur Authentifizierung benötigen eine zentrale Instanz, die über die notwendigen Informationen zur Prüfung der Authentizität verfügt. So müssen Paßwörter, aber auch Retinamuster in einer Datenbank gespeichert und mit der Nutzereingabe verglichen werden. Um eine Authentifizierung ohne zentrale Authentifizierungsstelle (wie etwa einen Kerberos- oder LDAP-Server) aufzubauen und somit der ersten der in Abschnitt 2.2 skizzierten Forderungen nachzukommen, wird

in der GSI (siehe Abs. 2.2.3) eine PKI, wie sie in Abschnitt 2.1.2 beschrieben wurde, verwendet.

Gegenseitige Authentisierung

Die gegenseitige Authentisierung („Mutual Authentication“) als Spezialfall der Authentisierung ist ein für vertrauenswürdige Kommunikation essentieller Vorgang, der sicherstellt, dass alle Kommunikationspartner von der Echtheit (also der Authentizität) des jeweils Anderen überzeugt sind. Während für die gegenseitige Authentisierung in nicht-PKI-basierten Umgebungen nur wenige Protokolle[2] existieren, wird diese Aufgabe in einer PKI durch die Nutzung von Zertifikaten deutlich erleichtert. Der in Abb. 2.2 illustrierte Vorgang läuft wie folgt ab: Unter der Voraussetzung, dass

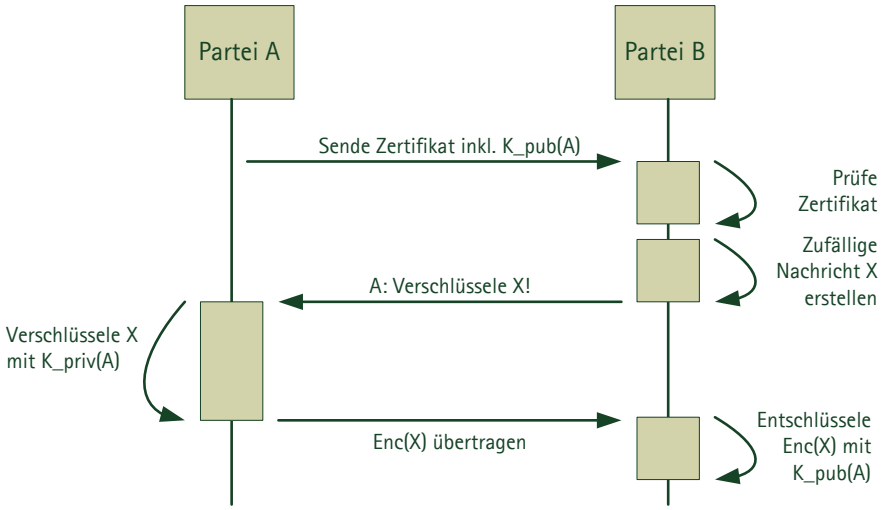


Abbildung 2.2: Gegenseitige Authentisierung

beide sich gegenseitig authentisierenden Parteien A und B über Zertifikate einer CA verfügen, der die jeweils andere Partei vertraut (im einfachsten Fall derselben CA), wird das folgende Protokoll zur Authentisierung angewendet:

1. A baut eine Kommunikationsverbindung zu B auf.

2. A sendet B sein Zertifikat, aus dem B den öffentlichen Schlüssel $K_{pub}(A)$ von A und die signierende CA erfährt.

[2] etwa das Interlock-Protokoll [RS84], das jedoch durch Bellovin und Merrit in [BM94] erfolgreich angegriffen wurde

3. B prüft die Signatur und Gültigkeit des Zertifikats.

4. B erstellt eine zufällige Nachricht X (etwa eine Zufallszahl) und sendet diese unverschlüsselt an A, mit der Aufforderung, X zu verschlüsseln.

5. A benutzt den zu seinem Zertifikat passenden privaten Schlüssel $K_{priv}(A)$, um X zu verschlüsseln und schickt X_{enc} zurück an B.

6. B verwendet $K_{pub}(A)$, um X_{enc} zu entschlüsseln. Wenn $Decrypt(X_{enc}) = X$, so ist der Nachweis der Identität von A erbracht.

7. Das Protokoll wird mit vertauschten Rollen wiederholt, um Bs Identität gegenüber A nachzuweisen.

2.1.4 AAI

Traditionelle Authentifizierungs- und Autorisierungsinfrastrukturen (AAI) sind häufig stark zentralisiert oder aber vollständig entkoppelt angelegt. Zentralisierte AAI zeichnen sich durch einen zentralen Dienst (wie etwa LDAP [Zei06] oder Kerberos [NYHR05] aus, der Authentifizierungsinformationen verwaltet und prüft, ob bei einem Authentifizierungsversuch die korrekten Credentials angegeben wurden. Ein typischer Anwendungsfall für zentralisierte AAI findet sich in der Netzwerkauthentifizierung in Unternehmensnetzen.

Bei lose oder gar nicht gekoppelten AAI sind alle Authentifizierungs- und Autorisierungsinformationen auf jeder Ressource vollständig vorhanden, so dass diese einen Authentifizierungsversuch eigenständig beurteilen kann. Das ist insbesondere dann notwendig, wenn eine Authentifizierung auch ohne Netzwerkverbindung erfolgreich durchgeführt werden soll – die Benutzer/Passwort-Liste `/etc/passwd` auf einem Unix-System wäre daher ein geeignetes Beispiel für eine dezentrale, nicht gekoppelte AAI. Diese ist zwar auch im Offline-Betrieb nutzbar, ihre Skalierbarkeit im Netzwerk- oder Clusterbetrieb läßt jedoch zu wünschen übrig. Schließlich müßten neue Nutzerkonten auf jeder Ressource einzeln eingerichtet werden, was im Allgemeinen als nicht wünschenswert erscheint.

Für die Authentifizierung und Autorisierung im Grid kommt jedoch weder eine zentralisierte noch eine vollständig entkoppelte AAI in Frage. Das mag widersprüchlich wirken, ist doch eines der wichtigsten Charakteristika des Grid-Computing die

Dezentralität, die bereits von Foster gefordert wurde. Das Spannungsfeld zwischen zentraler und stark entkoppelter Authentifizierung charakterisiert dieser in [Fos02].

> I suggest that the essence of the definitions above can be captured in a simple checklist, according to which a Grid is a system that: 1) coordinates resources that are not subject to centralized control ... (A Grid integrates and coordinates resources and users that live within different control domains – for example, the user's desktop vs. central computing; different administrative units of the same company; or different companies; and addresses the issues of security, policy, payment, membership, and so forth that arise in these settings. Otherwise, we are dealing with a local management system.)

Übers.: Ich schlage vor, dass die Essenz der obigen Definitionen in einer einfachen Checkliste aufgefangen werden kann, nach der ein Grid ein System ist, das: 1) Ressourcen koordiniert, die keiner zentralisierten [Zugriffs-]Kontrolle unterworfen sind... (Ein Grid integriert und koordiniert Ressourcen und Nutzer, die in verschiedenen administrativen Domänen zu Hause sind – zum Beispiel ein Desktoprechner des Nutzers gegenüber zentralen Recheneinheiten; verschiedene administrative Einheiten derselben Firma; verschiedene Firmen – und behandelt die Fragen der Sicherheit, Regulierung, Bezahlung, Mitgliedschaft und so weiter, die in solchen Umgebungen entstehen. Andernfalls handelt es sich um ein lokal administriertes System.)

Im derzeitigen Stand der Entwicklung im Grid-Computing wird diese Anforderung wie bereits im vorigen Kapitel skizziert durch eine PKI-basierte AAI erfüllt, deren universell vertrauensgebendes Element die Zertifikatsstelle ist. Diese CA ist jedoch kein Teil der Grid-Infrastruktur und demnach auch kein topologisch zentrales Element im Sinne einer zentralisierten AAI; eine Grid-PKI kann verschiedene CAs enthalten, denen jedoch von jedem Beteiligten vertraut werden muss. In der Praxis existiert für jedes an einem weltweiten Grid teilnehmende Land mindestens eine CA.

Autorisierungsinformationen für eine Grid-Infrastruktur werden ebenfalls dezentral auf jeder Grid-Ressource in Gestalt von grid-mapfiles (siehe Abschnitt 3.1.2) vorgehalten. Somit erfüllt eine typische Grid-Infrastruktur derzeit die Anforderungen an eine dezentrale AAI.

2.1.5 Autorisierung

Autorisierung (engl. Authorization, kurz AuthZ) bezeichnet im weitesten Sinne die Einräumung von Rechten an ein Subjekt durch eine Autorität. Die durch diese Rechte beeinflusste Ressource ist das Objekt der Autorisierung. So ist beispielsweise die Vergabe von Lese- oder Schreibrechten für eine Datei (Objekt) an einen Benutzer (Subjekt) in aller Regel von der Autorisierung durch den Besitzer der Datei (Autorität) abhängig. Jede Autorisierung wird sinnvollerweise nach einer vorangegangenen Authentifizierung erfolgen, da die Identität des Nutzers zunächst sichergestellt werden muss, bevor diesem Privilegien erteilt werden.

Neben der direkten Autorisierung in der Form „Subjekt X darf/darf nicht Aktion Y auf Objekt Z ausführen“ ist auch eine indirekte Autorisierung möglich, die bestimmte Eigenschaften des Subjekts berücksichtigt und anhand eines durch die Autorität definierten Regelsatzes eine Autorisierungsentscheidung fällt. Beide Autorisierungsmethoden werden in Grid-Infrastrukturen angewendet.

2.2 Grid-Computing

Die Idee eines globalen *Computing Grid*, also einer geographisch verteilten Infrastruktur für Supercomputing, ist die konsequente Fortführung der bereits in den 1970er Jahren begonnenen Ressourcenteilung. Angetrieben durch Entwicklungen in Forschungseinrichtungen, wurde das Paradigma, das zum Begriff des „Grid“ führte, von Ian Foster in einem oft zitierten Thesenpapier [Fos02] mit folgenden Forderungen geprägt:

1. *„The Grid coordinates resources that are not subject to a centralized control.“*
2. *„The Grid uses standard, open, general-purpose protocols and interfaces.“*
3. *„The Grid delivers non-trivial quality of service.“*

Die namensgebende Idee, nämlich ein weltweites Netzwerk für Rechenkapazität zu erschaffen, das analog zum Stromnetz (engl. „power grid“) jederzeit Rechenkapazität zuverlässig und praktisch beliebig skalierbar bereitstellt, mutet auch derzeit noch wie Zukunftsmusik an. Jedoch ist eine wichtige Teilidee bereits erreicht – der Wissenschaftler als Endanwender muss, um das Grid nutzen zu können, nicht jede Rechenressource kennen, sondern nur einen Zugang zu einem Scheduling-System

haben. Somit wäre es – um sich der Analogie zum Stromnetz erneut zu bedienen – nicht notwendig, ein Kraftwerk direkt anzusprechen, um Strom zu erhalten; lediglich der Kontakt zu einem Vermittler ist notwendig.

Auch die von Foster geforderte Dezentralität des Grid ist eingetreten; eine Vielzahl von Forschungsinstitutionen betreiben autarke Grid-Infrastrukturen, die untereinander technisch wie organisatorisch verbunden sind. Durch sog. „virtuelle Organisationen“ können Institutionen Teile ihrer Rechnerinfrastruktur, aber auch Personal für ein Grid bereitstellen.

Moderne Grid-Middleware-Systeme dienen jedoch längst nicht mehr nur der Vermittlung von Rechenkapazität und der Ausführung von Rechenjobs. Sie verwalten Datenressourcen, verfügen über Authentifizierungs- und Autorisierungsmechanismen sowie über Möglichkeiten zur Aggregation von Einzelaufgaben zu Workflows. Die Middleware sorgt hierbei für die Abstraktion von Rechen- und Datenressourcen und bietet dem Nutzer eine standardisierte Schnittstelle zur Grid-Infrastruktur.

In aktuellen Grid-Infrastrukturen existiert eine Vielzahl von Middlewares und Grid-Toolkits wie etwa:

- Das vom Argonne National Laboratory entwickelte *Globus Toolkit* [Fos05]
- Die am LHC entstandene und hauptsächlich in der EU verwendete Middleware *gLite* [LHP+04]
- UNICORE (Uniform Interface to Computing Resources) [Rom99], das vom Forschungszentrum Jülich als Supercomputing-Frontend entwickelt wurde und im Rahmen des D-Grid-Projekts als Grid-Middleware zum Einsatz kommt
- ARC (Advanced Resource Connector) [EGK+07], eine im skandinavischen Raum verbreitete Middleware

Allen Middlewares ist gemein, dass sie sich stark an der Forderung Fosters nach standardisierten, offenen und für verschiedene Zwecke benutzbaren Protokollen orientieren. Populäre Middlewares setzen für die Interaktion zwischen Rechenressourcen, Jobmanagement und Authentifizierungs-/Autorisierungs-Infrastrukturen oft auf webservice-basierte Kommunikation über die Transportprotokolle HTTP/HTTPS und verwenden an FTP angelehnte Protokolle für den Transfer großer Datenmengen. Das Gros der aktuell verwendeten Middlewares ist zudem quelloffen verfügbar und

Entwickler sind stets eingeladen, an der Verbesserung und Erweiterung mitzuarbeiten.

Betreiber einer Grid-Ressource können diese Ressource an beliebig viele Grid-Middlewares anbinden und somit einem großen Nutzerkreis technisch den Zugriff auf die Ressource ermöglichen. Im deutschen Grid-Infrastrukturprojekt D-Grid ist die Anbindung aller Ressourcen an die Middlewares Globus, UNICORE und gLite vorgesehen.

2.2.1 Virtuelle Organisationen

In Grid-Infrastrukturen basiert die Autorisierung von Nutzern nicht auf deren Zugehörigkeit zu einer realen Organisation (also ihrer Firmenzugehörigkeit oder universitärer Status), sondern auf ihrer Mitgliedschaft in einer oder mehrerer Communities, die ihrerseits wiederum autorisiert sind, das Grid zu nutzen. Insbesondere im D-Grid ist dieses Community-Konzept auch vom Mittelgeber umgesetzt worden; so werden wissenschaftliche Gemeinschaften etwa von Astronomen, Geoinformatikern oder Medizinern in speziellen Projekten gefördert.

Die Zugehörigkeit von Personen und Ressourcen zu einer oder mehrerer Communities wird durch virtuelle Organisationen (VO) abgebildet. Diese wurden erstmals von Foster in [FKT01] so bezeichnet:

> ...First, we review the „Grid problem“, which we define as flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources – what we refer to as virtual organizations.

Weiter führt Foster in derselben Publikation an, dass virtuelle Organisationen die Regeln für den gemeinsamen Zugriff auf Ressourcen selber festlegen:

> ...This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs.

Virtuelle Organisationen unterliegen also einer engen Zweckbindung und werden, sobald dieser Zweck erreicht ist, aufgelöst.

2.2.2 Globus Toolkit

Das Globus Toolkit ist eine Sammlung von Werkzeugen und Diensten zum Aufbau geographisch und organisatorisch verteilter Systeme und somit eine der „enabling technologies“ für moderne Grid-Infrastrukturen. Maßgeblich in seiner Entwicklung durch die „Globus Alliance“ getrieben, liegt das Globus Toolkit derzeit in seiner fünften Iteration vor und definiert Quasi-Standards für Protokolle und Verfahren im Grid.

Komponenten des Globus Toolkit

Die der vorliegenden Arbeit zugrunde liegende vierte Version des Globus Toolkit markiert eine Abkehr von den überwiegend auf proprietären Binärprotokollen basierenden Diensten der Vorgängerversionen hin zu Web Services. Aus Kompatibilitätsgründen wurden jedoch die nicht auf Web Services basierenden Komponenten beibehalten; einige Bestandteile des Globus Toolkit wie etwa der GridFTP-Server wurden zudem nicht mit WS-Schnittstellen versehen.

Die Komponenten des Globus Toolkit werden in fünf Hauptblöcke unterteilt:

- *Sicherheit*, also die Verwaltung von Authentifizierungs- und Autorisierungsmechanismen und -credentials
- *Datenmanagement* — die Bereitstellung zuverlässiger Transfer- und Speichermöglichkeiten für große Datenmengen
- *Jobmanagement* zur geordneten Abarbeitung und Orchestrierung einzelner Grid-Arbeitsschritte
- *Informationsdienste*, die eine automatisierte Dienstauffindung und das Accounting ermöglichen und
- Eine *Laufzeitumgebung* zur Entwicklung eigener, auf Globus basierender, Grid-Komponenten

Für das in dieser Arbeit vorgestellte Konzept sind insbesondere die Blöcke „Sicherheit“ und „Laufzeitumgebung“ relevant; Dienste aus Job- und Datenmanagement werden als Demonstrator für die Umsetzung herangezogen.

Im vorstehenden Architekturdiagramm ist das Auditing den Sicherheitsfunktionen zugeordnet, da es überwiegend sicherheitsrelevante Probleme bearbeitet. Es

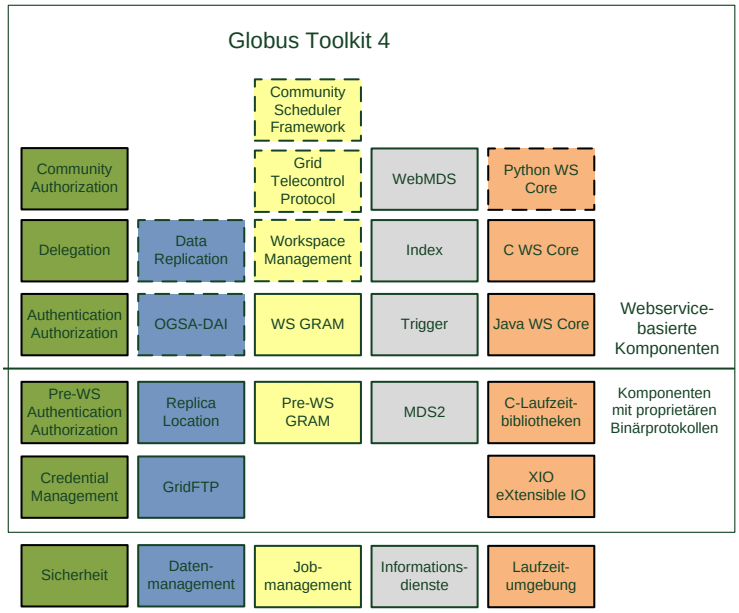


Abbildung 2.3: Architekturübersicht Globus Toolkit 4 (nach [Fos05]; gestrichelte Komponenten markieren *Technology Previews*)

könnte argumentiert werden, dass Auditing von Nutzercredentials als „Informationsdienst“ verstanden werden kann, jedoch dienen diese in GT4 ausschließlich der Information über zur Verfügung stehende Ressourcen und der Abrechnung.

Die für die vorliegende Arbeit relevanten Komponenten des Globus Toolkit werden im Folgenden kurz vorgestellt.

WS-Core

Der Web Service Core stellt im Globus Toolkit eine Referenzimplementation des Web Service Resource Framework (WSRF, siehe [CFF+04]), WS-Security und WS-Notification [GHM06] bereit. Die in C und Java (auszugsweise auch in Python) verfügbare Bibliothek ermöglicht die Entwicklung von Webservices, die auf statusbehafteten Ressourcen operieren. Die Verwaltung der Ressourcen gehört ebenso zum Funktionsumfang des WS-Core wie die automatische (De-)Aktivierung, Persistenzebenen und die Adressierung der Webservices. Der Java WS-Core wird mithilfe des Apache-Projekts Axis[3] implementiert; mit ihm entwickelte Webservices können in einem Tomcat Webservice-Container deployed werden.

[3] http://ws.apache.org/axis2/index.html

Globus Web Service Container

In Java implementierte und webservice-basierte Globus-Dienste können anders als ihre auf Binärprotokollen basierten Vorgänger nicht selbständig ausgeführt werden; sie benötigen einen sog. „Webservice Container“. Der Globus Container dient als „Gefäß“ und generische Serverumgebung für die Anwendungen. Er implementiert die notwendigen Basisfunktionen (also die Verwaltung von Netzwerkverbindungen, Datenhaltung und Zugriff auf Statusinformationen). Die Installation eines Web Service im Container wird gemeinhin als „Deployment“ bezeichnet. Typischerweise werden Globus-Webservices entweder in einem Tomcat-Applikationsserver[4] oder dem „Globus Container“ deployed. Dieser auf dem Apache-Projekt Axis basierende Webservice-Container implementiert neben zu Tomcat vergleichbaren Funktionen zusätzlich die Unterstützung für die GSI und kann somit „out of the box“ als sichere Umgebung für Web Services verwendet werden.

WS-GRAM

Der WS-GRAM (Web Service Grid Resource Allocation and Management) ist die Globus-Komponente, die Grid-Jobs vom Nutzer entgegennimmt und an ein Subsystem, meist einen Cluster-Scheduler wie Torque, weiterleitet. Die Bandbreite der Komplexität der angenommenen Rechenjobs ist sehr verschiedenartig; neben der Verwaltung von Ein- und Ausgabedaten übernimmt der GRAM auch das Job-Monitoring und die Signalisierung bei Statusänderungen.

Einer der zentralen Paradigmenwechsel zwischen verschiedenen Versionen des Globus Toolkit betrifft auch die Komponente GRAM: Wurde vor Version 4.0 noch ein Binärprotokoll für den Nachrichtenaustausch zwischen GRAM und Client verwendet, so schwenkten die Globus-Entwickler mit jener Version auf ein Webservice-basiertes Konzept, das auf dem Standard WSRF [CFF+04] beruht. Die nun als *pre-WS-GRAM* bezeichnete Komponente wurde im Globus Toolkit 4 zwar noch mitgeliefert, aber nicht mehr standardmäßig eingesetzt.

Mit Globus Toolkit 5 wurde – einem erneuten Paradigmenwechsel der Entwickler geschuldet – der webservice-basierte GRAM wieder durch eine verbesserte Version des *pre-WS-GRAM* ersetzt. Da diese Arbeit zu großen Teilen während der Lebensdauer von Globus Toolkit 4 konzipiert wurde und diese Version noch bis mindestens

[4] Tomcat-Projektseite: `http://tomcat.apache.org/`

2012 die primäre im D-Grid eingesetzte Globus-Version sein wird, wird im Folgenden zunächst der WS-GRAM genauer betrachtet.

Der WS-GRAM bietet nach außen WSRF-basierte Schnittstellen an und kapselt somit über ihn erreichbare Grid-Ressourcen. Seine primäre Aufgabe ist die Abstraktion verschiedener Cluster-Scheduler und -Scheduling-Mechanismen; der WS-GRAM enthält keine eigene Scheduling-Logik.

Bei der Abgabe von Grid-Jobs an einen WS-GRAM werden an verschiedenen Stellen Proxy-Credentials zur Authentifizierung und Delegation verwendet. Die Delegation von Credentials gewinnt insbesondere dann an Bedeutung, wenn Ein- und Ausgabedaten zwischen Compute- und Datenressource kopiert werden müssen (Stage In bzw. Stage Out).

GridFTP

Bei dem 2005 durch das Global Grid Forum (GGF) standardisierten GridFTP[5] handelt es sich um eine Erweiterung des File Transfer Protocol (FTP[6]), um die in Grid-Umgebungen notwendigen Kriterien der Zuverlässigkeit, Leistungsfähigkeit und Vertraulichkeit zu erfüllen. Die Erweiterungen des FTP-Standards beinhalten eine Erhöhung der Protokollperformance durch sichere „Third Party Transfers“, mehrere simultane TCP-Streams, Striping und Interleaving.Eine fehlertolerantere Implementierung von FTP ermöglicht u.a. den automatischen Neustart eines Transfers, sollten Probleme aufgetreten sein. Authentifizierung und Autorisierung werden durch die GSI erledigt; somit können Proxy Credentials als Login-Token verwendet werden. Durch die umfassende Unterstützung in vielen Grid-Middlewares (u.a. Globus, ARC, gLite) und eine Vielzahl von Clientprogrammen ist GridFTP das einzige relevante Protokoll zur Datenübertragung im Grid. Mit dem in Globus Toolkit 4 eingeführten RFT (Reliable File Transfer) steht ein webservice-basiertes Frontend zur Kontrolle von GridFTP-Verbindungen über WSRF-Webservices zur Verfügung.

2.2.3 GSI

Die *Grid Security Infrastructure* (GSI) [FKTT98] ist die in den meisten aktuellen Grid-Middlewares gebräuchliche Implementation von Sicherheitsfunktionen. Sie basiert auf einer PKI und erfüllt drei Hauptziele:

[5] `http://globus.org/toolkit/docs/2.4/datagrid/deliverables/C2WPdraft3.pdf`
[6] `http://tools.ietf.org/html/rfc959`

- Sichere, also authentische, unbestreitbare und vertrauliche Kommunikation zwischen allen Elementen eines Grids.

- Organisationsübergreifende Sicherheit, die ohne ein zentrales System zur Überwachung und Verwaltung von Rechten und Aktionen auskommt.

- Ein Mechanismus, der „Single Sign-On“ ermöglicht, um Aufgaben ohne weitere Nutzerinteraktion von mehreren Ressourcen und/oder an verschiedenen Orten ausführen zu lassen.

Die drei eingangs genannten Anforderungen der Authentizität, Unbestreitbarkeit und Vertraulichkeit bilden zentrale Voraussetzungen einer sicheren Infrastruktur. So muß jeder Kommunikationspartner verlässliche und verifizierbare Informationen haben, mit wem er gerade kommuniziert, denn ohne diese Authentizität kann eine Autorisierung auf zugriffsgeschützten Ressourcen nicht stattfinden. Vertraulichkeit, also die technische Beschränkung der Kommunikation auf den vorgesehenen Teilnehmerkreis, ist ebenso essentiell wie die Nichtbestreitbarkeit von Kommunikation – ohne diese beiden Sicherheitsmerkmale wäre die Prozessierung schutzwürdiger Daten im Grid ebenso wenig möglich wie die zuverlässige Zuordnung verbrauchter Ressourcen zu einem Nutzer oder einer Nutzergruppe.

Um authentische und vertrauliche Ende-zu-Ende-Kommunikation zu gewährleisten, wird im Globus Toolkit in der Regel Transport-Level-Security über TLS verwendet. Dieses weit verbreitete Protokoll baut aus Geschwindigkeitsgründen auf einen hybriden Ansatz aus symmetrischer und asymmetrischer Verschlüsselung.

Nachdem die Kommunikationspartner ihre Identität mittels des „Mutual Authentication“-Protokolls (siehe Abschnitt 2.1.3) nachgewiesen haben, wird über den so aufgebauten sicheren Kommunikationskanal ein symmetrischer Schlüssel, der „Session Key“, übertragen. Sämtliche weitere Kommunikation über den TLS-Kanal wird im Folgenden mit diesem verschlüsselt. Gegenüber der deutlich aufwendigeren asymmetrischen Kryptographie ergibt sich so ein deutlicher Geschwindigkeitsvorteil, was insbesondere in hoch belasteten Umgebungen (wie etwa einem GridFTP-Server, der Datentransfers mit Datenraten im Gbps-Bereich durchführen muss) die Gesamtleistung des Grid deutlich erhöht.

Kommunikationssicherheit in der GSI

Die GSI unterstützt eine Absicherung der Kommunikation durch Verschlüsselung und Signatur auf zwei verschiedenen Ebenen einer Kommunikation – der Transportebene (Transport Level Security) und der Nachrichtenebene (Message Level Security). Auf der Transportebene wird die gesamte Kommunikation verschlüsselt; die Verschlüsselung und Signatur auf Nachrichtenebene hingegen stellt die Vertraulichkeit der Nachricht sicher, indem nur ihr Inhalt, nicht jedoch Protokoll- und Metainformationen verschlüsselt werden. Bei der im Globus Toolkit 4 verwendeten, auf Web Services basierenden Kommunikation würde somit das XML-Gerüst um eine Nachricht im Klartext übertragen, die Nachricht selber aber verschlüsselt. In den Abbildungen 2.4 und 2.5 sind die Vorgänge in der Transport- und Nachrichtenverschlüsselung illustriert.

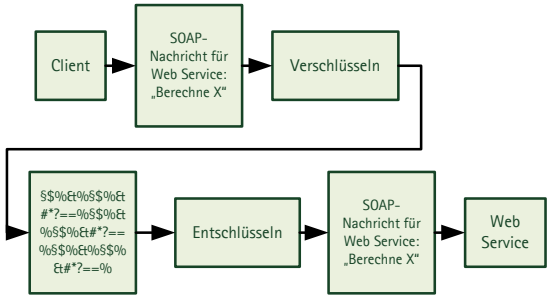


Abbildung 2.4: Verschlüsselung auf Transportebene (nach [SC06])

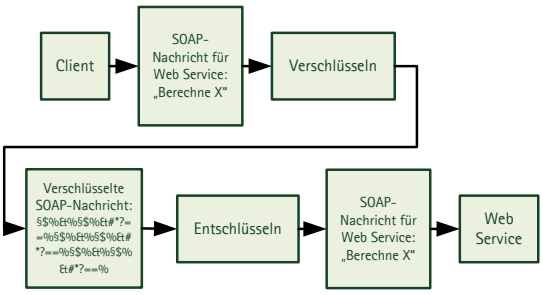


Abbildung 2.5: Verschlüsselung auf Nachrichtenebene (nach [SC06])

Beiden Kommunikationsmodi ist gemein, dass Integrität, Vertraulichkeit und Authentizität gewährleistet bleiben; sie unterscheiden sich jedoch in der Performance. Transport-Level Security bietet in der Regel eine höhere Leistung und wird daher

nicht nur in der GSI als Standardmethode zur Herstellung eines sicheren Kommunikationskanals genutzt.

Insgesamt existieren im Globus Toolkit drei verschiedene Modi zur Absicherung der Kommunikation:

- **GSI Secure Message** basiert auf dem WS-Security-Standard [NKMHB06],
- **GSI Secure Conversation** basiert auf dem WS-SecureConversation-Standard [NGG+07] und bietet durch die Etablierung eines *Security Context*, der für die gesamte Kommunikation aufrecht erhalten wird, eine bessere Leistung als GSI Secure Message. Delegation von Credentials wird nur mittels GSI Secure Conversation unterstützt.
- **GSI Transport** ist eine auf TLS basierende Implementation, bietet die beste Leistung, unterstützt aber keine Delegation.

Die von den einzelnen Sicherheitsmodi unterstützten Sicherheitsfeatures werden in Tabelle 2.1 miteinander verglichen. Für den Kontext dieser Arbeit sind lediglich

	GSI Secure Message	**GSI Secure Conversation**	**GSI Transport**
Basistechnologie	WS Secure Conversation	WS Security	TLS
Vertraulichkeit (Verschlüsselung)	✓	✓	✓
Integrität (Signatur)	✓	✓	✓
Anonyme Authentifizierung	✓		✓
Delegation	✓		
Eignung	Bei hohem Nachrichtenaufkommen	Bei niedrigem Nachrichtenaufkommen	Universell

Tabelle 2.1: Vergleich von TLS und MLS
Vergleich von Transport-Level und Message-Level Security (nach [SC06])

GSI Secure Conversation und GSI Transport relevant, da diese die üblicherweise verwendeten Sicherheitsschemata in Globus-Komponenten darstellen.

Autorisierung in der GSI

Die GSI erlaubt verschiedene Mechanismen zur direkten und indirekten Autorisierung von Nutzern und Ressourcen. Einige dieser Autorisierungsmechanismen werden im Folgenden kurz vorgestellt. Ohne vorherige Authentifizierung ist eine Autorisierung nicht möglich.

Gridmap und identitätsbasierte Autorisierung

Eine Möglichkeit der Autorisierung in einer GSI-basierten Infrastruktur ist die sogenannte „Gridmap“. Diese Datei, die auf jeder GSI-Ressource zu finden ist, enthält eine Zuordnung (Mapping) von Zertifikats-DNs auf ressourcenlokale Credentials (also meist Unix-Accountnamen) und bildet somit eine ACL (Access Control List) ab. Beim Zugriff auf eine Ressource prüft der GSI-Stack das vorgezeigte (Proxy-)Zertifikat und ordnet dem Zugreifenden die im grid-mapfile für den Zertifikats-DN eingetragenen lokalen Credentials zu. Ähnlich agiert die identitätsbasierte Autorisierung, die nur Nutzern mit einer bestimmten Identität den Zugriff erlaubt.

Rollenbasierte Autorisierung

Die in Abschnitt 2.2.1 vorgestellten virtuellen Organisationen können neben ihrer Funktion als Modell der rechtlich-organisatorischen Gegebenheiten in einer Grid-Infrastruktur zudem dazu dienen, die für Autorisierungsentscheidungen notwendigen Informationen zu liefern. Zunächst kann diese Entscheidung aufgrund der Mitgliedschaft des Subjekts in einer bestimmten VO gefällt werden; darüber hinaus besteht zudem für den Verwalter der VO die Möglichkeit, die Rolle des Subjekts innerhalb der VO auszudrücken. Aufgrund dieser Rolle kann dann mithilfe eines von der Autorität bestimmten Regelsatzes eine indirekte Autorisierungsentscheidung gefällt werden.

Um die Authentizität der Rollenbeschreibung eines VO-Mitglieds sicherzustellen, werden die entsprechenden Attribute in Form eines Attributzertifikats [FH02] ausgestellt, das vom Administrator der VO vermittels des „VO Management Service“ (VOMS) signiert wird. Durch diese Signatur, deren Validität mit den üblichen Mitteln geprüft werden kann, ist eine nachträgliche Veränderung der Attribute unmöglich.

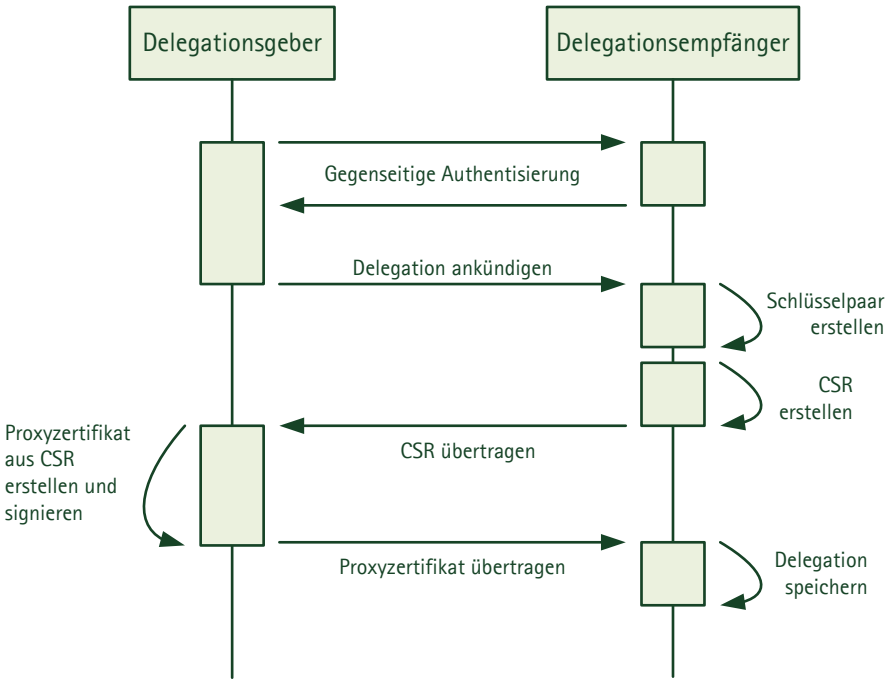


Abbildung 2.6: Ablauf der Credential-Delegation im Grid

Delegation

Ein Schlüsselmechanismus der Sicherheitsinfrastruktur im Grid ist die Delegation von Rechten durch Nutzer an Ressourcen. Sie ist unverzichtbarer Bestandteil der AAI des Grid und ermöglicht erst die Abgabe mehrschrittiger Aufgaben, die eine erfolgreiche Authentifizierung und ausreichende Autorisierung auf den beteiligten Ressourcen benötigt.

So ist bereits bei einem vergleichsweise einfachen Arbeitsablauf häufig eine Delegation von Rechten notwendig. Wenn etwa ein Nutzer einen Grid-Job von einer Rechenressource ausführen lässt, die Ein- und Ausgabedaten jedoch auf einer anderen Ressource vorgehalten werden, wird eine Delegation von Rechten notwendig. Die Rechenressource muß sich zunächst authentifizieren (was mittels der im folgenden Abschnitt skizzierten Single-Sign-On-Funktionalität geschieht) und dann eine ausreichende Autorisierung zum Abruf oder zur Speicherung von Daten auf der Speicherressource nachweisen. Diese Autorisierung kann jedoch nur erfolgen, wenn der Nutzer die notwendigen Rechte vorher an die Rechenressource delegiert hat.

Die Delegation von Rechten wird im Grid – ebenso wie die Mechanismen zum Single Sign-On – mittels Proxy-Zertifikaten implementiert. Dieser Vorgang läuft

stets über eine GSI-gesicherte Netzwerkverbindung ab und macht sich die Tatsache zunutze, dass von Proxy-Zertifikaten weitere Ableitungen gebildet werden können.

1. Die delegierende Stelle äußert gegenüber dem Empfänger ihren Delegationswunsch.

2. Der Delegationsempfänger erstellt ein Schlüsselpaar sowie einen Zertifikats-Request (CSR) basierend auf den Identitätsdaten des vom Delegationsgeber vorgezeigten Proxy-Zertifikats. Der CSR, nicht jedoch der zugehörige private Schlüssel, werden an den Delegationsgeber übertragen.

3. Der Delegationsgeber signiert den CSR mit dem geheimen Schlüssel des Proxyzertifikats, von dem die Delegation abgeleitet wird und überträgt es an den Delegationsempfänger.

Der Delegationsempfänger verfügt nun über ein gültiges Proxy-Zertifikat sowie den dazugehörigen privaten Schlüssel und kann sich somit gegenüber anderen Ressourcen im Grid als vom Zertifikatsinhaber beauftragte Stelle ausweisen.

Single Sign-On

Die wenigsten Grid-Infrastrukturen sind ohne sog. *Single Sign-On* sinnvoll nutzbar. Müsste der Nutzer bei jeder Interaktion mit einem Grid-Knoten erneut seine Zugangsdaten eingeben, wäre nicht nur die Benutzerfreundlichkeit, sondern auch eine der grundlegenden Ideen des Grid Computing nicht umsetzbar: Die Bereitstellung von Rechenleistung in einem stark dezentralisierten System ohne weiteres Zutun des Nutzers.

Wenn der Nutzer einen Rechenjob bei seiner Grid-Site abgegeben hat, werden in aller Regel Eingabedaten von einer Speicherressource auf die Rechenressourcen übertragen. Da diese Daten jeweils nur nach erfolgreicher Authentifizierung und Autorisierung zugänglich sind, müssen die notwendigen Zugangsdaten entweder vom Nutzer zum Zeitpunkt des Abrufs erneut eingegeben oder durch den Ressourcen-Manager-Dienst zwischengespeichert werden, um sie an die anfragende Ressource weitergeben zu können.

In der GSI werden zu diesem Zweck Proxyzertifikate genutzt, die gleichzeitig als Authentifizierungs- und als Autorisierungs-Hilfsmittel dienen. Mithilfe der ihm durch

den Nutzer zur Verfügung gestellten Proxy-Credentials kann der GRAM eine weitere Ableitung unterzeichnen, die dann der Rechenressource zur Authentifizierung gegenüber der Datenressource dient. So muss der Nutzer nicht allen Ressourcen im Voraus explizit Credentials übermitteln und das Missbrauchsrisiko wird gegenüber einer Übermittlung etwa von Benutzername und Passwort im Klartext verringert.

Mangels wirksamer feingranularer Autorisierungsmechanismen in Globus-basierten Grids wird die mit dem Single Sign-On einhergehende Delegation der Nutzerrechte meist nach dem Prinzip „alles oder nichts“ vorgenommen – wer im Besitz eines gültigen Proxycredentials ist, kann alle Aktionen ausführen, die auch der Nutzer selber unter Vorlage seines Nutzerzertifikats ausführen dürfte.

2.2.4 Proxy-Credentials

Um wichtige Sicherheitsfunktionen wie etwa Single-Sign On und Delegation von Rechten (siehe jew. Abschnitt 2.2.3) bereitzustellen, werden in der GSI *Proxy-Zertifikate* verwendet. Diese in RFC 3820 [TWE+04] beschriebenen Zertifikate werden – im Gegensatz zu den von einer akkreditierten CA signierten End-Entity- oder Nutzerzertifikaten – vom Nutzer selber mittels eines geeigneten privaten Schlüssels signiert. Dieser private Schlüssel kann im Falle eines „Proxy-Zertifikats ersten Grades“ der zum Nutzerzertifikat gehörende Schlüssel sein; es ist ebenfalls möglich, ein Proxy-Zertifikat mit dem privaten Schlüssel eines weiteren Proxy-Zertifikats zu unterschreiben.

Durch diese Verkettung ergibt sich eine lediglich indirekte Verbindung zwischen dem CA-signierten Zertifikat eines Nutzers und einem Proxy-Zertifikat, die es einer Grid-Ressource oder einem anderen Nutzer dennoch erlaubt, die Authentizität der Signaturen bis zum betreffenden Nutzer zurückzuverfolgen. Somit ergibt sich für ein Proxy-Zertifikat, dass es (in den Grenzen seiner Laufzeit) als ebenso vertrauenswürdig angesehen wird wie das zugehörige Nutzerzertifikat.

Die Gültigkeit von Proxyzertifikaten kann in mehreren Domänen beschränkt werden. Das RFC 3820 sieht zunächst eine maximale **Gültigkeitsdauer** von 1.000.000 Sekunden (etwa 11 Tagen), also eine Beschränkung in der Zeitdomäne, vor – in der Praxis hat sich eine Laufzeit von 12 Stunden für Proxyzertifikate als „good Practice“ für viele Anwendungen herauskristallisiert. Durch die wesentlich geringere Laufzeit werden Proxyzertifikate für einen Angreifer weniger nützlich, denn selbst wenn ihm ein Schlüsselpaar (etwa durch mangelhafte Absicherung auf einer Grid-Ressource)

in die Hände fällt, bietet sich nur ein kurzes Zeitfenster, in dem dieses Schlüsselpaar Missbraucht werden kann.

Zudem haben Arbeiten wie das Konzept von Piger et al. zur **Beschränkung der Rechte** von Proxy-Zertifikaten durch den Nutzer ([PGGK08], [Pig08]) Möglichkeiten aufgezeigt, das Missbrauchspotential dieser Credentials weiter einzuschränken. Keines dieser Konzepte hat jedoch die Entwickler der wichtigsten Grid-Middlewares überzeugt, so dass bei der weiteren Betrachtung davon ausgegangen werden muß, dass Proxyzertifikate bis auf die kürzere Laufzeit dieselben Berechtigungen innehaben wie die sie ausstellenden Zertifikate. Selbst wenn ein wirksames Mittel zur Einschränkung der Gültigkeit von Proxy-Zertifikaten existierte, würde dies die Notwendigkeit für Auditing der Credentials nicht verringern, denn ein Missbrauchspotential wäre noch immer gegeben.

Im Gegensatz zu CA-signierten Zertifikaten können Proxyzertifikate nicht zurückgezogen (revoked) werden. Diese Einschränkung ergibt sich aus der Tatsache, dass ein Proxyzertifikat bei der CA, die dessen Ausstellerzertifikat signiert hat, unbekannt ist und demzufolge auch nicht in ihrer CRL verzeichnet werden kann. Dem Nutzer, der das Proxyzertifikat signiert hat, steht zudem keine standardisierte Möglichkeit zur Verfügung, eine eigene CRL zu pflegen, die von Grid-Ressourcen abgefragt werden könnte. Konzepte wie das von Luna in seiner Dissertation beschriebene ([LMM05]) zielen darauf ab, diese Einschränkung aufzuheben – Abschnitt 1.5.5 enthält weitere Informationen zu diesen Ansätzen.

Da ein Zertifikat, auch ein Proxyzertifikat, als Authentifizierungsmittel untauglich ist (schließlich enthält es nur öffentliche Informationen), hat sich für die Kombination aus einem Proxyzertifikat und dem dazugehörigen privaten Schlüssel der Terminus „Proxy Credential“ eingebürgert, der - in Ermangelung einer griffigen deutschen Entsprechung - im Folgenden genutzt werden soll.

Modellierung der Beziehungen zwischen digitalen Zertifikaten

Eine anschauliche Modellierung für die Beziehungen zwischen Zertifikaten ergibt sich, indem Familien- oder Verwandtschaftsbeziehungen zwischen ihnen angenommen werden. Diese Modellierung erleichtert es, präzise Aussagen über die Relation von Zertifikaten zu machen, gleichzeitig werden sperrige und wenig nachvollziehbare Begrifflichkeiten jedoch vermieden. Die Entsprechungen realer Verwandtschaftsbeziehungen bei X.509-Zertifikaten sollen im Folgenden kurz demonstriert werden.

Jedes digitale Zertifikat wird mithilfe des privaten Schlüssels eines weiteren, des Urheberzertifikats signiert. Dessen CA-weit eindeutige Seriennummer wird in jedem von ihm signierten Zertifikat vermerkt (seit X.509 Version 2, siehe [ITU05], Anhang 1). Bei konventionellen X.509-PKIs wird in der Regel der private Schlüssel des CA-Zertifikats zur Signatur aller EECs verwendet. Somit sind sie in der im folgenden verwendeten Modellierung „Geschwister“ voneinander, da sie direkt vom selben Urheberzertifikat abstammen. Diese Beziehung wird somit als Eltern-Kind-Beziehung modelliert; ein CA-Zertifikat hat in dieser Terminologie keine Eltern (es wurde mit seinem eigenen privaten Schlüssel signiert).

Zur Erstellung von Proxyzertifikaten kann hingegen, wie im vorigen Abschnitt erläutert, sowohl ein EEC als auch ein anderes Proxyzertifikat verwendet werden. Es kann somit sowohl Eltern als auch Geschwister und Kinder haben.

2.3 Die Grid-Landschaft in Deutschland und Europa

Grid-Computing wird in Deutschland insbesondere durch die Initiative „D-Grid“ vorangetrieben. Sie umfaßt die größte Zahl an Ressourcenbetreibern und beteiligten Community-Projekten in Deutschland. Das Projekt „European Grid Initiative“ , das aus dem EU-Projekt EGEE (Enabling Grids for E-sciencE) hervorging, kooperiert mit dem D-Grid und verfolgt das Ziel einer universell nutzbaren europäischen Grid-Infrastruktur.

2.3.1 D-Grid

Das „D-Grid“ wird als gemeinsame Initiative von Wissenschaft und Wirtschaft seit 2005 vom BMBF gefördert und hat den Aufbau einer nationalen Grid-Infrastruktur in Deutschland zum Ziel. Das 2008 erfolgreich beendete „D-Grid Integrationsprojekt 1“ (DGI-1) schuf eine Infrastruktur, die seitdem als Basis für Projekte aus verschiedenen wissenschaftlichen Communities und Public-Private-Partnerships dient. Die über ganz Deutschland verteilten Standorte des D-Grid sind über das deutsche Wissenschaftsnetz breitbandig miteinander verbunden.

Im Kontext dieser Arbeit dient das D-Grid als praktisches Beispiel einer massiv verteilten Grid-Infrastruktur und der mit ihr verbundenen Probleme und Schwächen.

Einige durch die Arbeit in der Administration des RRZN-eigenen Grid-Clusters erworbene Erkenntnisse fließen als motivierende Faktoren ein. Zusätzlich wird der Nachweis der Gültigkeit des Auditing-Konzepts anhand ausgewählter Projektpartner im D-Grid erbracht.

Architekturelle Besonderheiten

Wie in jedem anderen nationalen Grid wurden Leistungsmerkmale des D-Grid an den Anforderungen der Projektpartner und Nutzer ausgerichtet, was zu einigen Besonderheiten geführt hat. Diese – sofern für den Kontext dieser Arbeit relevant – sollen hier kurz aufgeführt werden. Für weitere Informationen steht das D-Grid Betriebskonzept online [Fie09] zur Verfügung.

Drei Middlewares

Da die D-Grid-Infrastruktur von verschiedenen wissenschaftlichen Communities genutzt wird, müssen Ressourcenbetreiber Zugänge für drei verschiedene Middlewares bzw. Grid-Toolkits zur Verfügung stellen:

- gLite
- Globus Toolkit (derzeit in Version 4.0.x)
- UNICORE

Obgleich im Kontext dieser Arbeit nur eine Auditing-Lösung für das Globus Toolkit entwickelt wurde, ist die Entscheidung, drei Middlewares zu unterstützen, dennoch relevant, schafft sie doch auf Grid-Ressourcen mehr Angriffsfläche für Attacken von böswilligen Dritten, die Proxy-Credentials und damit Grid-Ressourcen übernehmen wollen (siehe auch Kapitel 3).

Zentrale Komponenten

Einige Grid-Komponenten werden im D-Grid zentral betrieben und sollen daher nicht von den Ressourcenbetreibern oder Communities in Eigenregie betrieben werden. Dazu gehören:

- ein zentraler MyProxy Credential Repository Service

- die Systeme zum Monitoring und zur Ressourcenverwaltung (GRRS, MDS und WebMDS)
- die Nutzer- und VO-Verwaltung VOMRS
- ein zentraler User-Interface-Server (UI)
- sowie middleware-spezifische Systeme zur Job-Submission und -Verwaltung

Der zentrale Betrieb sicherheitsrelevanter Komponenten, insbesondere des MyProxy und der UI-Services, wirft Fragen auf, die in Kapitel 3 weiter erörtert werden.

Zwei CAs

In Deutschland sind derzeit zwei CAs durch die EU Grid Policy Management Authority (EUGridPMA, [EUG06]) akkreditiert: Die CA des DFN e.V. und die vom Karlsruhe Institute of Technology (KIT) betriebene GridKA-CA. Um eine eindeutige Identifizierbarkeit eines Nutzers mittels des Zertifikats-DN gewährleisten zu können, wird das Organisations-Feld zur Unterscheidung der jeweiligen Grid-CA genutzt (`O=GridGermany` für DFN-CA sowie `O=GermanGrid` für KIT-CA). Es ist trotz zwei zuständiger EUGridPMA-akkreditierter CAs in Deutschland problemlos möglich, Nutzer eindeutig anhand ihres Zertifikats-DNs zu identifizieren.

2.3.2 EGI und NGI-DE

Die European Grid Initiative (EGI) hat sich nach eigener Aussage das Ziel gesetzt, „eine europaweite Grid-Infrastruktur zu schaffen und zu unterhalten“ [Eur11]. Das Nachfolgeprojekt der Initiative „Enabling Grid for E-SciencE“ (EGEE) dient dabei als zentrale Stelle zur Standardisierung und Umsetzung technischer Entwicklungen für die Teilnehmer, die NGIs (National Grid Infrastructure). Strategische Entscheidungen von EGI-weiter Bedeutung werden von einem Beirat getroffen, der sich aus Vertretern der NGIs und Vertretern der Nutzer-Communities zusammensetzt.

In Deutschland werden Aktivitäten des D-Grid mit denen des NGI-DE – also der deutschen nationalen Grid-Initiative – koordiniert und beide Initiativen agieren in enger Kooperation. So ist die Nutzung der D-Grid Compute-Ressourcen für Zwecke des NGI-DE möglich. Die Gauß-Allianz fungiert als Projektleitung für die NGI-DE.

Im Kontext dieser Arbeit ist die Relevanz von NGI-DE insofern gegeben, als dass die zu schaffenden Infrastrukturen in ihrer Skalierung nicht nur landes-, sondern

unter Umständen auch europaweit dimensioniert werden müssen, sofern das Auditing von Proxy-Credentials vom EGI gewünscht wird.

2.3.3 Andere Initiativen

Das US-amerikanische *TeraGrid*[7] ist eine der einflußreichsten Grid-Initativen weltweit und umfaßt elf Projektpartner, darunter das Argonne National Laboratory und die Universität Chicago. Neben einer umfangreichen Infrastruktur für Compute-Jobs und Datenspeicherung sind Projektpartner aus dem TeraGrid federführend in der Entwicklung des Globus Toolkit, der technischen Grundlage des in dieser Arbeit vorgestellten Konzepts.

NorduGrid[8] ist ein Zusammenschluß aus fünf skandinavischen Forschungsinstituten. Die Grid-Middleware ARC [EGK+07], die auf Globus Toolkit 2 basiert, wird im Rahmen der NorduGrid-Initiative entwickelt. Die Initiative agiert auch als Anbieter für eine CA, die EUGridPMA-akkreditierte Zertifikate für Nutzer aus Dänemark, Finnland, Norwegen und Schweden ausstellt.

2.4 Auditing

2.4.1 Begriffsabgrenzung

In aktuellen und vergangenen Veröffentlichungen wurde der in dieser Arbeit einen hohen Stellenwert einnehmende Begriff des *Auditing*, aber auch die verwandten Begriffe Logging, Tracking, Tracing sowie in den Bereich des Auditing hineinragende Themenkomplexe wie Monitoring und Accounting nicht immer einheitlich verwendet, weswegen eine grundsätzliche Begriffsfindung notwendig wird, um die in dieser Arbeit gebräuchlichen Termini klar zu definieren.

Der Begriff *Audit* (etym.: lat. *audire* - hören, anhören) wird heute häufig als Paraphrase für eine Prüfung spezifischer Vorgänge oder Prozesse gebraucht. In der Finanz- und Wirtschaftswelt wird ein Audit mit Wirtschaftsprüfungen, aber auch Konformitätsprüfungen zu gebräuchlichen Workflow- und Unternehmensstandards (etwa der Kontrolle der Sicherheitsstandards nach ISO 27001 [Int08a] im Rahmen eines *Certification Audit*) verbunden. In der Informations- und Kommunikations-

[7] TeraGrid-Projektwebseite: `https://www.teragrid.org/`
[8] Nordugrid-Projektwebseite: `http://www.nordugrid.org/`

technologie werden unter Anderem Sicherheitsprüfungen (*Security Audit*) und Softwareprüfungen (*Software audit*) vorgenommen, die unter dem Oberbegriff *Audit* subsummiert werden.

Zusätzlich wird das zielgerichtete Sammeln von Informationen, die für einen Audit notwendig sind, statt sperriger Begriffe wie „Auditing-Datenerhebung“ ebenfalls kurz mit Auditing bezeichnet. Die erhobenen Daten sind in der Literatur oft als „Auditing-Informationen“ oder „Auditing-Daten“ bezeichnet.

2.4.2 Auditing im Kontext dieser Arbeit

Da der Begriff „Auditing“ in der Literatur uneinheitlich besetzt ist und verschiedene Anforderungen und Umsetzungen existieren, ist zunächst zu ergründen, welche Informationen für den Zweck der vorliegenden Arbeit erhoben und verarbeitet werden sollen und welchen Zweck diese Verarbeitung erfüllt.

Im Kontext des D-Grid-Projekts MediGRID [KBB+09] wurde bereits 2006 der Bedarf an grid-weitem Auditing zur Sicherstellung der Vertraulichkeit sensitiver medizinischer Daten geäußert. So wurde während des 1. D-Grid Security-Workshops durch Jürgen Falkner das Auditing als das Sammeln von „Nutzungsdaten [..], von wem in welcher Form auf sicherheitsrelevante Daten zugegriffen wurde“ [Fal06] definiert. Die sicherheitsrelevanten Daten sind in diesem Zusammenhang vom Grid-Nutzer beigebrachte Daten, die unter anderem Patienteninformationen oder andere sensible medizinische Informationen enthalten können. Falkner fordert, Datenzugriffe, Systemzustände sowie Policy-Verstöße zu erfassen, um Auskunftspflichten gegenüber dem Nutzer bzw. dem Patienten nachkommen und sicherheitsrelevante Vorfälle rekonstruieren zu können.

Diese weit gefaßte Definition des Auditing-Begriffs kann die Kontrolle von Datenzugriffen, aber auch die detaillierte Protokollierung der Rahmenbedingungen und Ergebnisse von Compute-Jobs beinhalten.

In der vorliegenden Arbeit wird dieser Forderung Rechnung getragen, da der Zugriff auf ein im Grid-Kontext besonders sicherheitsrelevantes Datum, nämlich das Proxy Credential auditiert wird. Da jeder Datenzugriff und jede Nutzung einer Grid-Ressource ein gültiges Credential voraussetzt, kann mittelbar durch Proxy-Auditing auch der Zugriff auf Daten- und sonstige Ressourcen auditiert werden.

Auditing von Grid-Jobs

In einer 2009 von M. Stratmann verfassten und vom Autor dieser Dissertation betreuten Arbeit [Str08] wurde das Auditing von Compute-Jobs untersucht. Das – durch Middleware-Bestandteile wie dem GRAM Audit Logging [The07] implementierte – Auditing von Grid-Jobs dient dieser Untersuchung zufolge zweierlei Zweck:

1. In den heterogen aufgebauten Grid-Umgebungen großer nationaler Infrastrukturen ist die Nachvollziehbarkeit von Job-Ergebnissen oft nicht nur anhand der Jobbeschreibung (etwa in RSL oder JSDL) zu gewährleisten; zu viele Rahmenparameter wie Versionen und Verhalten von Betriebssystem, Bibliotheken und Anwendungssoftware variieren zwischen Rechenknoten verschiedener oder gar desselben Cluster. Dieser durch Nebenläufigkeiten bedingte Nichtdeterminismus kann bei der Auswertung von Experimenten eine schwer zu identifizierende und reproduzierende Fehlerquelle bilden.

2. Eine genaue Verfolgung gerechneter Jobs erleichtert dem Anbieter die Belegbarkeit (Non-Repudiation), wann und von wem Rechenressourcen konsumiert wurden und welche Kosten ggf. dadurch entstanden. Hier spielt das Auditing dem betriebswirtschaftlichen Accounting zu.

Auditing von Datenzugriffen

Die in einer Grid-Infrastruktur schutzbedürftigsten Elemente sind die Daten der Nutzer. Unprozessierte Rohdaten enthalten experimentell gewonnene Resultate, die meist nur schwer oder gar nicht wiederhergestellt werden können, wenn sie verloren gehen. Von Ausgabedaten können wichtige wissenschaftliche Ergebnisse abgeleitet werden, die über Gedeih und Verderb eines Projekts entscheiden können.

Besonders schutzbedürftig sind die in der Medizin erhobenen Daten, da sie durch ihren Personenbezug unter besondere datenschutz- und standesrechtliche Bestimmungen fallen; ihre Vertraulichkeit gegenüber Dritten muss stets und unbedingt gewährleistet sein.

Unter diesem Gesichtspunkt liegt es nahe, die Zugriffe auf Daten separat zu auditieren. Dies kann geschehen, indem die für Datenzugriffe genutzten Dienste, also GridFTP, dCache [Fuh04] und OGSA-DAI [AAB+05] mit Schnittstellen versehen werden, die jeden Zugriff an einen Auditing-Dienst melden. Ein solches Daten-Auditing stellt somit eine Erweiterung der entsprechenden Serverdienste dar.

Im Rahmen dieser Arbeit wurde jedoch auf eine dezidierte Implementation verzichtet, da das Auditing von Proxy-Credentials alle Datendienste bereits mit einschließt, da diese die Schnittstellen und Kommunikationsprotokolle der GSI implementieren. Somit kann jeder versuchte Zugriff auf eine Datenressource nachgehalten werden.

2.4.3 Audit Records

Ein einzelnes, nicht korreliertes Audit-Datum wird in dieser Arbeit als *Audit Record* (also „Audit-Datensatz“) bezeichnet; hier handelt es sich um ein atomares Tupel, das eine Nutzung eines Proxy-Credentials auf einer Grid-Ressource anzeigt und für sich genommen meist wenig Aussagekraft besitzt. Für jeden GSI-Authentifizierungs-Vorgang und jede Delegation wird mindestens je ein Audit Record durch die Auditing-Infrastruktur erhoben.

In einem Audit Record wird erfaßt, wer (Zertifikats-DN des Proxy-Zertifikats) wann (Timestamp der Nutzung) was (Nutzung für Authentifizierun oder Delegation) gemacht hat - somit werden alle notwendigen Informationen für die spätere Nachvollziehbarkeit gesammelt. Detaillierte Informationen über Entwurf und Implementierung der Audit Records sind in den Abschnitten 4.4.3 und 5.2 zu finden.

2.4.4 Audit Trails

Korreliert man verschiedene Audit Records anhand eines verbindenden Merkmals, etwa eines zeitlichen oder personellen Zusammenhangs, so entsteht ein Informationsgewinn gegenüber der singulären Betrachtung (Emergenz). Diese Korrelation kann in der in dieser Arbeit vorgestellten Auditing-Infrastruktur auf verschiedene Arten erfolgen:

- Aggregation anhand zeitlicher Abfolge
- Audit Records, die von einem bestimmten Quell-Host stammen
- Audit Records, die demselben Nutzer (anhand des Zertifikats-DN) zuzuordnen sind
- Audit Records, die dieselbe Nutzungsart oder denselben Nutzungszweck des zu auditierenden Proxy Credentials beinhalten

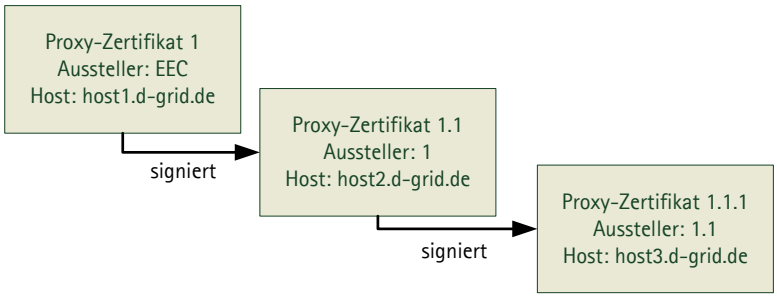


Abbildung 2.7: Audit-Trail des Proxy-Zertifikats mit der Seriennummer 1.1.1

- Audit Records, die dieselbe Zertifikats-Seriennummer betreffen

Für diese Arbeit stellt es sich als zweckmäßig dar, Auditinginformationen anhand der ihnen zugrundeliegenden Credentials zu gruppieren. Hier leistet die Seriennummer des Zertifikats wertvolle Dienste.

Wie in 2.2.4 detailliert erläutert wird, enthält jedes Zertifikat eine eindeutige Seriennummer und die Seriennumer desjenigen Zertifikats, von dem es abgeleitet (d.h. mit dessen privatem Schlüssel es unterschrieben) wurde.

Es ist nun möglich, von einem Nutzerzertifikat (EEC) bis zu jedem Proxy-Zertifikat eine Kette der ableitenden Zertifikate zu bilden und somit – da ein Proxy-Zertifikat nebst privatem Schlüssel auf der Grid-Ressource verbleiben soll, auf der es erzeugt wurde – eine lückenlose Kette der zu einem Credential vorliegenden Audit-Records zu bilden. Diese Kette wird im Folgenden als *Audit Trail* bezeichnet.

2.4.5 Rahmenbedingungen des Auditing

Das Auditing in Computersystemen – hier speziell im Zusammenhang mit den eher traditionellen Mechanismen des Logging und Accounting – wird vielfach durch Normen und Regelungen festgelegt. Ein Artikel von Meints und Thompson [MT07] fasst die konkreten Anforderungen prägnant zusammen und bezieht sich explizit auf die ISO/IEC-Normen 27001 [Int08a] und 27002 [Int08b], die das „Monitoring zum Erkennen unautorisierter Aktivitäten zur Datenverarbeitung“ durch „Audit Logging“ fordern, um die „(interne Auditierbarkeit technischer Systeme und Komponenten sicher[zu]stellen“.

Konkrete Vorschläge, wie die genannten „Audit Logs“ aufgebaut sein sollen, sind in der Norm ISO/IEC 27002 ebenso enthalten wie der Hinweis, dass die Manipula-

tion, Löschung und Überschreiben von Audit-Daten durch technische Maßnahmen, wie etwa eine revisionssichere Archivierung, verhindert werden muss.

2.4.6 Würdigung von Auditing in Sicherheitsstandards

Insbesondere in der kommerziellen IT sind Sicherheitsstandards und Zertifizierungen ein essentielles Qualitätskriterium für ausgelagerte („outsourced“) Computersysteme. Ohne nachgewiesene Einhaltung von Richtlinien wie die ITIL-Prozeßbeschreibungen oder ISO/IEC 27001 ist für Firmen eine Nutzung von Rechenressourcen Dritter im Rahmen von Grid oder Cloud Computing oft nicht möglich. Im folgenden Abschnitt wird untersucht, inwiefern Logging oder Auditing in den wichtigsten relevanten Standards abgedeckt sind und somit auch für die Betreiber von Grid-Infrastrukturen zu einer zwingenden Voraussetzung für die Nutzung durch kommerzielle Dritte werden.

BSI-Grundschutzkataloge

Für in Deutschland betriebene IT-Anlagen sind die Empfehlungen des Bundesamts für Sicherheit in der Informationstechnik (BSI) in den BSI-Grundschutzkatalogen [Bun] relevant, die sich in Bezug auf Logging und Auditing an ISO/IEC 27001 orientieren, jedoch zusätzlich detailliertere Maßnahmenkataloge enthalten. In Punkt M 2.64 („Kontrolle der Protokolldateien“) wird angeregt, dass Logdateien regelmäßig von dafür bestimmten und autorisierten Personen ggf. mit Hilfe spezieller Werkzeuge zu überprüfen sind. Laut Abschnit M 2.110 („Datenschutzaspekte bei der Protokollierung“) unterliegen Protokolldateien, insbesondere, wenn sie personenbezogene Daten beinhalten, den gesetzlichen Regelungen – in Deutschland also insbesondere denen des Bundesdatenschutzgesetzes (BDSG). Somit ist ihre Nutzung nur zweckgebunden zulässig.

Diese Zweckbindung ist durch die Sicherstellung des ordnungsgemäßen Betriebs einer EDV-Anlage regelmäßig gegeben (nach §14 Abs. 4 und § 31 BDSG [Ver90]); es ist jedoch zu untersuchen, inwiefern das Auditing von Proxy Credentials (die einen direkten Personenbezug anhand des DN ermöglichen) der Zweckbindung nach BDSG unterliegt. Eine ausführlichere Würdigung dieser datenschutzrechtlichen Implikationen von Proxy Auditing findet sich in Abschnitt 4.6.

ISO27001

Der internationale Standard ISO27001 „Information technology - Security techniques - Information security management systems - Requirements“ [Int08a] greift zwei Aspekte des Auditinggedankens auf: Die Prüfung von sicherheitsrelevanten internen Vorgängen steht im Vordergrund der in Abschnitt 6 des Standards erläuterten „Internal ISMS audits“ (die Abkürzung ISMS steht für Information Security Management System), während das eingangs bereits erwähnte „Audit Logging“ einen Kontrollmechanismus (Abschnitt A.10.10) darstellt, der im Rahmen eines Monitoring auf unautorisierte Zugriffe zu implementieren ist.

Die regelmäßige Prüfung der im Unternehmen implementierten Sicherheitsmechanismen soll im Rahmen interner Audits durch Firmenmitarbeiter durchgeführt werden und sicherstellen, dass regulatorische Bedingungen, aber auch die Kriterien der firmeneigenen Sicherheitsrichtlinie erfüllt sind. Auch die Effektivität und Korrektheit der Umsetzung von Sicherheitsmechanismen wird in einem derartigen Audit überprüft – Prüfer sollen neutral agieren und nicht eigene Arbeiten überprüfen.

Einer der im Rahmen eines solchen Audits zu prüfenden Mechanismen ist das Audit Logging, dessen Funktion folgendermaßen beschrieben wird: „Audit logs recording user activities, exceptions, and information security events shall be produced and kept for an agreed period to assist in future investigations and access control monitoring.“ [Int08a].

Alle sicherheitsrelevanten Ereignisse sollen also aufgezeichnet und für eine bestimmte Zeitspanne aufbewahrt werden, um bei zukünftigen Ermittlungen als Beweismittel dienen zu können. Die Speicherung soll, so die Bestimmungen in ISO27001 weiter, auf eine nicht manipulierbare Art und Weise erfolgen. Diese Definition von „Audit Logging“ repräsentiert den Auditing-Gedanken, wie er in dieser Arbeit vorherrscht.

Kapitel 3

Sicherheit und Auditing in Grid-Infrastrukturen

Im vorliegenden Kapitel erfolgt eine Analyse der Sicherheits- und Bedrohungslage in wissenschaftlich genutzten verteilten Systemen am Beispiel der nationalen deutschen Infrastruktur „D-Grid“. Auf dieser Analyse aufbauend, werden verschiedene konkrete Bedrohungsszenarien skizziert, die deutlich machen, dass Nutzer konkreten Gefahren ausgesetzt sind, die mit dem Sicherheitsmodell des Grid nicht gebannt werden können.

Der Hauptbetroffene der vorgestellten Bedrohungen ist der Grid-Nutzer. Die Implikationen für den Nutzer und mögliche Lösungsansätze werden im letzten Teil dieses Kapitels beschrieben.

3.1 Analyse der aktuellen Sicherheitslage im D-Grid

In der Initiative „D-Grid“, die vom Bundesministerium für Bildung und Forschung gefördert wird, haben sich wissenschaftliche Institutionen zusammengeschlossen, die gemeinsam die deutsche nationale Grid-Infrastruktur betreiben. In Abs. 2.3.1 wurden einige architekturelle Besonderheiten dieses föderierten Grids angeführt; deren Sicherheitsimplikationen sollen im Folgenden untersucht werden.

3.1.1 Authentifizierung per X.509 PKI

Das D-Grid setzt zur Authentifizierung von Nutzern auf eine X.509 PKI mit einem End-Entity-Zertifikat für jeden Teilnehmer (Nutzer oder Host). Zertifikate einer durch die EUGridPMA akkreditierten CA werden auf Ressourcen des D-Grid als Authentifizierungsmerkmale akzeptiert.

3.1.2 Autorisierung

VO-Mitgliedschaften

Der Zugang auf eine D-Grid-Ressource setzt zwingend die Mitgliedschaft in mindestens einer der virtuellen Organisationen des D-Grid voraus. Diese umfassen VOs für Infrastrukturprojekte, für wissenschaftliche Communities, aber auch für Lehrzwecke. An einigen Standorten ist ein Grid-Zugang für Studenten möglich, sofern eine Mitgliedschaft in der entsprechenden VO besteht.

Eine VO-Mitgliedschaft muss grundsätzlich beim jeweiligen VO-Manager abgesegnet werden und wird mittels eines webbasierten Vorgangs über den vom Forschungszentrum Jülich betriebenen VO Management & Registration Service (VOMRS) beantragt. Voraussetzung für die Mitgliedschaft ist ein von einer EUGridPMA ausgestelltes End-Entity-Zertifikat.

Zugriffsschutz per grid-mapfile

Unabhängig von einer Mitgliedschaft in einer D-Grid-VO können Ressourcenbetreiber Nutzer, denen sie den Zugriff auf ihre Ressourcen gestatten möchten, manuell in das lokale „Grid-Mapfile“ eintragen. Diese Eintragung ergibt insbesondere zu

Testzwecken oder für Mitarbeiter, die nicht einer bestimmten Grid-Community zugeordnet werden sollen, Sinn. Zugangsvoraussetzung ist jedoch auch hier ein gültiges Zertifikat.

3.1.3 Kommunikation und Netzwerksicherheit

Transportverschlüsselung

Alle Kommunikation zwischen D-Grid-Standorten (und der Großteil interner Kommunikation) findet über GSI-Schnittstellen statt und ist somit praktisch immer mit SSL bzw. TLS verschlüsselt. Somit ist das Risiko, dass unbeteiligte Dritte die Kommunikation abhören oder verändern können, bei angenommener ausreichender Schlüssellänge und -Sicherheit zu vernachlässigen.

Sobald jedoch nicht mehr gewährleistet ist, dass die zum Verschlüsseln einer Kommunikation verwendeten Schlüssel sicher sind (z.B. weil ein Proxy-Credential unerlaubt dupliziert wurde), können Dritte die Kommunikation zwischen Sites abhören und manipulieren. Mithilfe gestohlener Credentials kann mitgeschnittener Datenverkehr vom Angreifer auch nachträglich noch entschlüsselt werden; ein aktiver Man-in-the-Middle-Angriff ist hierfür nicht notwendig.

Firewall

Das D-Grid-Betriebskonzept [Fie09] sieht den Betrieb von Firewalls als Bestandteil einer sicheren Grid-Infrastruktur vor. Jede Grid-Site soll an ihrer Netzwerkgrenze zum Internet durch geeignete Firewallregeln unautorisierte Zugriffe auf interne Netzwerkdienste unterbinden. Die in [VG07] ausgesprochenen Empfehlungen beinhalten jedoch insbesondere für die Kommunikation mit GRAM, WS-GRAM und GridFTP-Servern die Freischaltung großer Portbereiche.

Zudem schützt auch eine sicher konfigurierte Firewall nicht gegen das in Abschnitt 3.4 skizzierte Bedrohungsszenario – hier finden Angriffe über explizit für Zugriffe von aussen freigeschaltete Ports und Dienste statt.

3.2 Nachvollziehbarkeit von Handlungen und Kommunikation

Ein wichtiges Kriterium für die Betriebssicherheit verteilter Infrastrukturen ist neben der technisch-organisatorischen Absicherung auch die Nachvollziehbarkeit jeglicher Aktivitäten in einer solchen Umgebung. Gerade recht lose gekoppelte und weltweit verteilte Grid-Infrastrukturen können nur dann verläßlich sein, wenn die Interaktion von Nutzern und Komponenten stets nachvollziehbar ist, also im Nachhinein eindeutig reproduziert und interpretiert werden kann. Dieses Kriterium ist insbesondere dann von Belang, wenn die Nutzer anhand ihres Ressourcenverbrauchs bemessene Zahlungen für die Grid-Infrastruktur zu leisten haben.

Jede aktuelle Grid- und Cloud-Middleware verfügt über entsprechende Mechanismen zur korrekten Verbuchung von Ressourcenverbrauch und Nutzeraktionen. Aufgrund ihres primären Einsatzzwecks, eine Grundlage für Accounting und Billing zu schaffen, sind diese Mechanismen jedoch nicht für eine nutzerzentrische Auditierung gedacht und auch nicht für eine solche geeignet. Die für das Logging und Accounting zuständigen Komponenten in Globus und gLite wurden in Abschnitt 1.5.7 erläutert; sie erlauben jedoch keine Erkennung und Einordnung mißbräuchlicher Nutzung.

3.3 Betriebssystemsicherheit

Die im D-Grid zusammengeschlossenen Rechencluster werden von den teilnehmenden Institutionen in Eigenregie betrieben. Diese sind insbesondere auch für die Betriebssystemsicherheit selber verantwortlich. Das Dokument zur D-Grid-Referenzinstallation sieht keine besonderen Maßnahmen zur „Härtung“ des Betriebssystems vor.

Die D-Grid-Cluster werden ausschließlich mit dem Betriebssystem Linux betrieben; die verwendete Linux-Distribution (also die Bündelung von Linux-Kernel, System- und Hilfsprogrammen zu einem vollwertigen und benutzbaren Betriebssystem) ist in der Regel eine Version von Scientific Linux[1].

Allen Nutzern des D-Grid wird durch das „Grid-Mapfile“ (siehe auch Abschnitt 3.1.2) anhand des Distinguished Name ihres Zertifikats ein Unix-Nutzer zugeordnet. Jobs und Dateizugriffe auf den Grid-Knoten werden mit den Berechtigungen dieses

[1] Projektseite Scientific Linux: `http://www.scientificlinux.org`

Nutzers ausgeführt. Somit kann jeder Nutzer des D-Grid auf jedem Grid-Knoten beliebige Aktionen mit den Unix-Nutzerprivilegien ausführen.

Die Sicherheit des Betriebssystems Linux hängt zu großen Teilen von der Sicherheit des Linux-Kernels ab. Dieser gilt als eines der größten und komplexesten Softwareprojekte weltweit und wird von einem Team mehrerer Hundert Entwickler verwaltet und weiterentwickelt. Trotz der bei dieser Entwicklung angewandten größten Sorgfalt schleichen sich bisweilen Fehler in den Kernel ein, die von einem geschickten Angreifer genutzt werden können, um die eigenen Nutzerprivilegien auf einem derart verwundbaren Rechner zu erhöhen. Da Kernelprozesse unter Linux stets vom privilegierten Nutzer `root` ausgeführt werden, ergibt sich durch einen erfolgreichen „Kernel-Exploit“ somit meist eine Erhöhung der Nutzerprivilegien zu ebendiesem Nutzer. Damit kontrolliert der Angreifer das komplette System.

Die Administratoren der im D-Grid eingesetzten Systeme werden durch das „Computer Emergency Response Team“ (CERT) des Deutschen Forschungsnetzes e.V. bei der Pflege ihrer Cluster insofern unterstützt, als dass kritische Sicherheitsprobleme in allen für das D-Grid relevanten Komponenten direkt nach Bekanntwerden vom DFN-CERT weitergemeldet werden – die Grid-Administratoren sind also über neue „Exploits“ in der Regel schnell informiert.

Da jedoch viele Sicherheitsprobleme von unabhängigen Forschern mit oft zweifelhaften Intentionen entdeckt werden, ist es nicht unwahrscheinlich, dass ein zur Privilegienerhöhung ausnutzbarer Fehler in einer Grid-Komponente zwar einer ausgewählten Benutzergruppe bekannt ist, einer breiteren Öffentlichkeit und insbesondere den Herstellern der betroffenen Software jedoch nicht zur Verfügung steht. Derlei Sicherheitslücken können natürlich weder programmatisch behoben noch durch den Administrator geflickt werden, so dass ein Restrisiko beim Betrieb einer an das Internet angeschlossenen Ressource stets eingegangen werden muss.

Wird eine Sicherheitslücke bekannt, erhöht sich am Tag der Veröffentlichung, dem „0 Day“ (Zero Day, Tag Null) das Gefahrenpotential immens. So ist nun nicht nur ein geschlossener Kreis von Crackern über die Möglichkeit informiert, sich erhöhte Privilegien auf einem Linux-Rechner zu verschaffen, sondern eine breite Öffentlichkeit von unter Umständen böswilligen Dritten. Gleichzeitig steht oft noch keine Möglichkeit zur Verfügung, das Problem zu beheben, da der Softwarehersteller ebenfalls vor vollendete Tatsachen gestellt wurde.

In einem solchen Fall gilt für D-Grid-Ressourcenbetreiber, dass ihre Ressourcen

de facto von jedem im D-Grid angemeldeten Nutzer übernommen werden können. Zu diesen Nutzern können auch Studenten zählen, die – etwa für ein Seminar über Grid-Computing – Zugriff auf das D-Grid erhalten haben[2].

Nach Veröffentlichung eines Software-Patches für die betroffene Komponente müssen alle Rechenknoten im D-Grid im Rahmen kurzfristiger Wartungen aktualisiert werden. Ob ein System auf dem aktuellsten Stand ist, wird jedoch derzeit nicht durch das D-Grid-Projekt geprüft. Es ist daher zu vermuten, dass einzelne D-Grid-Komponenten auf einem veralteten Softwarestand aufbauen und daher angreifbar sind. Diese schwächsten Glieder in der Sicherheitskette können einem Angreifer ein Sprungbrett für weitere Angriffe auf das gesamte D-Grid bieten. Die Sicherheit des Gesamtsystems D-Grid ist also nur so groß wie die des unsichersten beteiligten Systems. Gleichzeitig sind die Umsetzungen der vorgeschriebenen Sicherheitskonzepte innerhalb der deutschen Grid-Infrastruktur sehr heterogen.

3.4 Angriffsszenarios

Das in diesem Kapitel exemplarisch für nationale Grid-Infrastrukturen behandelte D-Grid nutzt die in Kapitel 2 erläuterten Mechanismen einer PKI und die Grid Security Infrastructure zur Authentifizierung und Autorisierung. Die Authentifizierung mittels Proxy Credentials sowie die Delegation sämtlicher Nutzerrechte birgt jedoch Risiken, die bei einem erfolgreichen Angriff gegen das Grid den möglichen Schaden vervielfachen können.

Angriffe von außen

Grid-Infrastrukturen nutzen, insbesondere bei der Föderation von Ressourcen über Organisations- und räumliche Grenzen hinweg, das Internet als Transportnetz. Sie exponieren damit also zumindest Teile der Infrastruktur nach außen. Angriffe über das Internet durch nicht als Teilnehmer einer Grid-Infrastruktur eingetragene Dritte werden als Angriffe von außen betrachtet. Auf eine Grid-Infrastruktur werden sowohl ungerichtete, breit gestreute Angriffsversuche als auch punktgenau geplante und ausgeführte Attacken ausgeführt werden.

[2]Für Lehrveranstaltungen und studentische Labore wurde eine spezielle VO im D-Grid geschaffen.

Ungerichtete Angriffe, die häufig von Botnetz-Betreibern zur Vergrößerung ihrer „Herde“ durchgeführt werden, werden immer mit vollautomatisierten Werkzeugen durchgeführt. Diese nutzen Sicherheitslücken im Betriebssystem sowie in häufig verwendeten Serveranwendungen aus, um Schadsoftware auf einem übernommenen System zu installieren. Gegen derlei Angriffe schützen oft bereits einfache Maßnahmen wie etwa die Verlegung von häufig genutzten Serverdiensten (HTTP, SSH, GSI-SSH etc.) auf untypische TCP-Ports. Selbst nach einer erfolgreichen Penetration eines Servers wird ein solcher Angreifer nur selten von seinem Schema abweichen – zu hoch ist die schiere Zahl der von ihm angegriffenen Maschinen, als dass er sich persönlich auf einem übernommenen Rechner umschauen würde. Demnach ist von ungerichteten Angriffen aus dem „Darknet“ für das Grid wenig Gefahr zu erwarten.

Die Interessen der Organisatoren ungerichteter Angriffe sind meist strikt kommerzieller Natur und zielen auf die direkte Verwertung der übernommenen Server und Dienste für Zwecke des Angreifers ab. Zu diesen Zwecken zählt hauptsächlich die massenhafte Verbreitung unerwünschter Werbung per E-Mail sowie das Hosting der in den Spam-Mails beworbenen Webseiten. Die Entwickler moderner Botnets wie etwa Zeus [Zha08], Kraken, Bredolab) haben hierzu komplexe und auf hohe Redundanz und Sicherheit vor Rückverfolgung ausgelegte Protokolle entwickelt (wie etwa den „Fast Flux“-Mechanismus zum Loadbalancing von Webseiten durch viele Bots). Häufig wird die CPU-Leistung der gecrackten Server auch eingesetzt, um Einheiten der virtuellen Währung „BitCoin“[3] zu errechnen [Hyp11]; hochperformante Rechencluster zählen für diesen Zweck zu den begehrtesten Zielen.

Einen erfolgreich durch Botnetz-Betreiber übernommenen Server zu entdecken, in der Regel einfach und durch den Einsatz eines Intrusion Detection Systems (IDS) (wie es etwa im D-Grid-Projekt GIDS [HgE+10] für die Anforderungen eines modernen Grid angepaßt wird) weitgehend automatisierbar.

Zu Vorgehen, technischen Hilfsmitteln und eingesetzter Software bei ungerichteten Angriffen auf Internet-Sites (und damit auch Grid-Sites) existiert umfassende und hochdetaillierte Literatur. Die Erforschung und Bekämpfung von Botnetzen stellt mittlerweile einen etablierten und stetig wachsenden Zweig des Security Engineering dar.

Zielgerichtete Angriffe sind zwar deutlich seltener zu erwarten, allerdings gehen die Angreifer hier insgesamt planvoller und vorsichtiger zu Werke. Da wenige erfolg-

[3] Projektwebseite: `http://www.bitcoin.org/`

reiche Angriffe auf Grid-Infrastrukturen bekannt sind, sind Motivation und Vorgehen nicht anhand empirischer Daten zu ergründen. Es ist jedoch im Sicherheitsmanagement nicht grid-basierter verteilter Strukturen zu beobachten, dass Angreifer großen logistischen und organisatorischen Aufwand nicht scheuen, wenn sie Industriespionage oder -sabotage verüben wollen. So wurde der Computerwurm „Stuxnet“ [FMC11] speziell mit dem Ziel entwickelt und verteilt, eine bestimmte Industrieanlage der Firma Siemens zu sabotieren, wie sie u.a. in Kraftwerksteuerungen eingesetzt wird.

Es ist also davon auszugehen, dass ein ambitionierter Angreifer, der zielgerichtet eine verteilte Infrastruktur ausspionieren und/oder manipulieren möchte, über beträchtliche Ressourcen verfügt und so dem „Verteidiger“ stets einen Schritt voraus ist. Somit muss auch angenommen werden, dass es einem solchen Angreifer regelmäßig gelingen wird, durch eine Lücke in einer Komponente in eine Grid-Infrastruktur einzudringen.

Das Betriebskonzept des D-Grid sieht vor, dass sämtliche Sites im D-Grid von einer Firewall geschützt werden, die Angriffe von außen wirksam abwehrt. Dieser Schutz gilt jedoch nicht für solche Komponenten, die dezidiert zur Kommunikation mit dem Endnutzer bestimmt sind. Dazu zählen sowohl User-Interface-Rechner als auch – je nach typischem Arbeitsablauf in der jeweiligen VO – Clientrechner, von denen aus Jobs submittiert und Ein-/Ausgabedaten verwaltet werden.

3.4.1 Angriffe von innen

Grid-Infrastrukturen werden, insbesondere im D-Grid, von Industriepartnern genutzt, die – ähnlich wie Forschungsinstitute – häufig in direkter Konkurrenz zueinander stehen. Industriespionage ist hier eine reale Gefahr, die in Sicherheits-Workshops und entsprechenden Dokumenten durch Industrievertreter regelmäßig geäussert wurde.

Spitzenforschung ist zudem – abseits von der interinstitutionellen Zusammenarbeit in Forschungsprojekten und wissenschaftlichen Vorhaben – ein ebenso wettbewerbsorientiertes Feld wie die Entwicklung neuer Produkte in der Industrie. Forschungsförderung durch Regierungs- und Industriefonds wird zunächst an diejenigen Institute vergeben, die sich durch hochwertige Veröffentlichungen und wissenschaftliche Erkenntnisse auszeichnen. In vielen wissenschaftlichen Bereichen wie etwa der Pharmakologie konkurrieren zudem mehrere Forscherteams um ein- und dasselbe wissenschaftliche Ergebnis. Es ist somit auch im Bereich der Wissenschaft durchaus

nicht weit hergeholt, dass die Forschungsergebnisse einer wissenschaftlichen Institution für Dritte so interessant sein können, dass diese vor dem Einsatz unlauterer Mittel bei deren Beschaffung nicht zurückschrecken.

Die Nutzung eines Grid als ver-, aber auch zwischen verschiedenen Projekten geteilter Infrastruktur birgt das Risiko, dass Forschungsergebnisse durch einen erfolgreichen Angriff in die Hände anderer Forschungsinstitutionen gelangen, deren Mitarbeitern zwar die Nutzung des Grid, nicht aber der Zugriff auf die betreffenden sensitiven Daten gestattet ist.

Bei möglichen Tätergruppen für Insider-Angriffe auf eine Grid-Infrastruktur lohnt sich erneut ein Blick auf die diesbezüglichen Überlegungen im traditionellen Security Engineering. So könnte ein verärgerter Projektmitarbeiter die Projektdaten manipulieren oder ein Student sich mittels eines nicht ganz legalen Kunstgriffs mehr Rechenzeit und Speicherplatz im Grid verschaffen wollen.

Allen möglichen internen Angreifern auf eine Grid-Infrastruktur ist jedoch gemein, dass sie über einen gültigen Account in diesem Grid verfügen, ihre Privilegien jedoch erhöhen wollen. Sie müssen somit nicht wie externe Angreifer eine Lücke in von außen erreichbaren Services einer Grid-Komponente finden, sondern können im Schutze ihres legalen Accounts nach Kernel- oder anderen Sicherheitslücken suchen. Es ist somit anzunehmen, dass diese Angreifergruppe sich ähnlich verhalten wird wie ein externer Angreifer. Sie wird versuchen, Kontrolle über ein oder mehrere gültige Proxy-Credentials zu erlangen, um diese für ihre eigentliche Absicht zu nutzen.

Vorgehen eines externen Angreifers

Wie zuvor bereits erwähnt, müssen von aussen erreichbare Grid-Ressourcen, insbesondere „interaktive Knoten“[4] oder UI-Server, als mögliche Einfallstore für unbefugte Dritte gesehen werden. Bei einer Komponente, die für Endnutzer nur per SSH oder GSI-SSH erreichbar ist, wird der Angreifer zunächst unter Ausnutzung einfach erratbarer Nutzerpaßwörter (Bruteforcing) oder über ein anderes Verfahren[5] unprivilegierten Zugriff auf das UI-System erlangen, um danach – etwa, indem er eine Sicherheitslücke im Linux-Kernel ausnutzt – Administrator-Privilegien zu erlangen.

[4] also Cluster-Nodes, die per GSI-SSH erreichbar sind, um Grid-Jobs leichter testen zu können

[5] Hat der Angreifer zuvor den Arbeitsplatzrechner eines Grid-Nutzers erfolgreich angegriffen und z.B. ein Programm zum Mitschneiden von Tastatureingaben installiert, so kann er die SSH-Zugangsdaten dieses Nutzers und möglicherweise sogar die Passphrase seines End-Entity-Zertifikats abfangen und zum Login auf einem UI-Server verwenden.

Sodann steht der UI-Rechner unter seiner Kontrolle und weiterer Missbrauch ist ohne Weiteres möglich.

Insbesondere kann der Angreifer nun auf dem UI-Server hinterlegte Credentials anderer Nutzer für seine Zwecke benutzen, da diese – im Gegensatz zu End-Entity Credentials – in aller Regel nicht über eine Passphrase gesichert sind. Das Missbrauchspotential potenziert sich somit, denn mit Hilfe gültiger Proxy-Credentials kann ein Angreifer mit den Rechten eines legitimen Grid-Nutzers auftreten und auf alle Ressourcen und Dienste zugreifen, die auch diesem zur Verfügung stehen.

Erlaubt eine Grid-Infrastruktur Zugriffe nicht nur auf UI-Server, sondern auch etwa auf Job- und Datenmanagement über das Internet, so vergrößert sich die Erfolgswahrscheinlichkeit für einen Angriff mit jedem nach aussen exponierten Dienst.

3.4.2 Weitere Angriffsszenarios

In einigen Grid-Middlewares werden sog. „Pool Accounts“ verwendet. Die Zuordnung von Endnutzer zu Unix-Account (wie in 3.1.2 beschrieben) wird hier nicht mit n:n, sondern mit n:m (mit m<n) vorgenommen, so dass auf einen Pool-Account mehrere Nutzer kommen. Dieses Account-Pooling (wie es etwa bei gLite möglich ist) stellt die Vertraulichkeit der vom Nutzer eingebrachten Daten mittels Aufräumroutinen fest, die nach Abschluss jedes Jobs die Arbeitsverzeichnisse leeren. Teil dieser Daten ist häufig auch eine Ableitung des zur Abgabe des Jobs verwendeten Proxy-Credentials, da auch ein Cluster-Knoten in vielen Fällen über eine gültige Delegation verfügen muss. Schließlich muss auch zur Laufzeit eines Jobs noch die Möglichkeit bestehen, Eingabedaten nachzuladen oder Ausgabedaten zu speichern, ohne auf die middleware-spezifischen Mechanismen des Stage-In[6] und Stage-Out[7] zurückzugreifen. Delegationen werden ebenfalls im Arbeitsverzeichnis abgelegt; meist als unverschlüsselte Textdatei.

Schlägt die „Garbage Collection“ – etwa durch einen nicht korrekt terminierten Grid-Job oder einen Absturz des Job-Schedulers – fehl, kann ein gültiges Proxy Credential in die Hände eines Unbefugten gelangen, der zufällig nach dem Inhaber dieses Credentials einen Job auf dem betreffenden Cluster-Knoten ausführt.

```
tar -czf \
```

[6]Laden von Eingabedaten vor Ausführung eines Grid-Jobs; Teil der Job-Definition in JDL/RSL

[7]Kopieren der Ausgabedaten eines Grid-Jobs auf eine Datenressource; Teil der Job-Definition in JSDL/RSL

```
2    /tmp/ser.tgz \
     /home/username/.globus/persisted/*/DelegationResource/*.ser
4  scp \
     /tmp/ser.tgz \
6    tmp@badhost:/tmp/ser.tgz
```

Listing 3.1: Automatische Duplizierung von Credentials per Cron-Job

Auf manchen Betriebssystem-/Middleware-Kombinationen können Nutzer zudem Aufgaben in einen Planer zur zeitversetzten Ausführung eintragen („at“ oder „Cron“). So kann ein Angreifer mittels eines regelmäßig ausgeführten „Cronjobs“ auf einem Clusterknoten neu im Arbeitsverzeichnis gespeicherte Proxy-Credentials automatisch duplizieren lassen und für seine Zwecke verwenden. Ein kurzes Skript, das im Rahmen eines solchen Cronjobs regelmäßig die serialisierten Credential-Dateien aus dem überwachten Heimatverzeichnis auf einen entfernten Server kopiert, ist in 3.1 aufgeführt.

3.5 Folgen erfolgreicher Angriffe

Sobald ein Angreifer die Kontrolle über ein gültiges Proxy-Zertifikat nebst dem zugehörigen unverschlüsselten privaten Schlüssel erlangt hat, kann er während der Laufzeit dieses Credentials alle Aktionen ausführen, zu denen auch der legitime Besitzer berechtigt ist. Er kann diesem somit auf mehrere Arten schaden.

3.5.1 Mißbräuchliche Ressourcennutzung

Der Angreifer kann das Proxy-Credential dazu benutzen, Rechenzeit und Speicherplatz – deren Nutzung in Grid-Infrastrukturen durch Accountingsysteme erfasst wird – zu verbrauchen und somit dem legitimen Nutzer oder seiner Heimatinstitution finanziellen Schaden zufügen. Das Zeitfenster für diesen Missbrauch ist durch die Gültigkeit des Proxy-Zertifikats begrenzt; in Grid-Middlewares, die das Renewal, also die Verlängerung eines auslaufenden Proxy-Credentials, erlauben, kann dieser Zeitrahmen jedoch beträchtlich sein.

Die mißbräuchliche Ressourcennutzung wird vom Nutzer regelmäßig erst dann erkannt werden, wenn das für die Grid-Ressourcen zuständige Billing erfolgt; also etwa durch eine deutlich zu hohe Rechnung am Ende eines Abrechnungszeitraums.

Zu diesem Zeitpunkt kann die mißbräuchliche Nutzung jedoch nicht mehr verhindert werden und ein finanzieller Schaden ist in jedem Fall entstanden. Ob dieser den Grid-Betreiber oder den Nutzer trifft, hängt von den vertraglichen Regelungen zwischen beiden ab.

3.5.2 Spionage

Eine häufige Folge erfolgreicher Angriffe auf Computernetze ist der Zugriff auf geheime Informationen, die dem Angreifer einen finanziellen oder intellektuellen Vorteil verschaffen können. Industriespionage ist seit den Anfängen organisierten Crackings eine der wichtigsten Angriffsmotivationen (wie etwa beim „KGB-Hack“ [ALMS89] Mitte der 1980er Jahre). In jüngerer Zeit wurden insbesondere aus China vermehrt Angriffe auf die Netze von Forschungseinrichtungen, aber auch Firmen durchgeführt. Im Rahmen der „Operation Aurora“ wurden 2009 namhafte Firmen wie Google, Juniper, Adobe, aber auch die Großbank Morgan Stanley angegriffen und auf sensible Informationen wie etwa die E-Mails chinesischer Dissidenten, unautorisiert zugegriffen.

Nutzer von Grid-Infrastrukturen kommen aus verschiedenen wissenschaftlichen und kommerziellen Bereichen, in denen teilweise hochsensible Informationen verarbeitet werden. Gelängen etwa die im Grid gespeicherten Entwicklungsmodelle eines großen Autoherstellers in fremde Hände, könnte dies einen immensen finanziellen Verlust bedeuten. Ähnlich verhält es sich bei den kommerziellen Geodaten und -algorithmen, die im Rahmen des D-Grid-Projekts „GDI-Grid“ von den Projektpartnern im Grid gespeichert wurden.

Auch Forschungsprojekte können sensible und sicherheitsrelevante Daten enthalten – etwa in der Kernforschung, aber auch und insbesondere in der Informatik, etwa in der Entwicklung von „Dual Use“-Projekten, die sich für eine zivile und militärische Nutzung eignen.

Für das Opfer eines Spionageangriffs ist der Schaden oft schwer bezifferbar – während die eigenen Projekte durch einen solchen Angriff nicht zwingend behindert werden, erlangen die Angreifer jedoch Informationen, die ihnen auf legalem Wege nicht zugänglich gewesen wären und ihre Wettbewerbsposition verbessern oder neue Sicherheitsfragen aufwerfen können.

Der entstehende Schaden durch derlei unautorisierte Duplikate interner Daten wird durch ein Beispiel aus dem Jahre 2009 deutlich. Damals wurden durch ei-

nen Angreifer interne E-Mails des Klimaforschungsinstituts der University of East Anglia publiziert [Kul09], die umfangreiche Manipulationen und andere Interna offenlegten und für eine dauerhafte Diskreditierung der beteiligten Wissenschaftler in ihrer Community sorgten.

3.5.3 Sabotage von Daten

Mit einem gültigen Proxy-Credential kann der Angreifer sämtliche Daten und Software, die der legitime Nutzer auf Grid-Ressourcen vorhält, manipulieren. Zum Einen kann er durch das Löschen von Ein- oder Ausgabedaten eine Denial-of-Service-Situation herbeirufen, er kann jedoch auch auf subtilere Weise die Integrität der Daten verletzen. So könnte ein Angreifer ein Forschungsexperiment sabotieren, indem er Ausgabedaten unauffällig manipuliert und somit falsche Ergebnisse erzeugt.

Die Sabotage und unerlaubte Duplizierung von Daten im Grid ist – im Gegensatz zur widerrechtlichen Ressourcennutzung – nur selten durch das Accounting im Grid feststellbar. Daher ist es für das Opfer einer solchen Manipulation noch schwieriger als bei der oben erwähnten Missbrauchsvariante, einen Angriff zu erkennen.

3.5.4 Manipulation von Grid-Jobs

Durch den Zugriff auf die Job-Scheduling-Dienste der Grid-Infrastruktur kann ein Angreifer mit einem gültigen Proxy-Credential nicht nur (wie oben beschrieben) Ressourcen missbräuchlich nutzen, sondern auch die Ressourcennutzung durch den legitimen Nutzer verhindern, indem er bereits laufende Jobs abbricht oder durch die massenhafte Abgabe fehlerhafter Jobs eine „Denial of Service“-Situation provoziert.

3.5.5 Angriffe gegen weitere Grid-Ressourcen

Hat der Angreifer ein oder mehrere gültige Proxy-Credentials von einer Grid-Komponente kopiert, kann er diese während ihrer Laufzeit[8] verwenden. Eines seiner Hauptziele wird jedoch sein, die unerlaubte Nutzung des Grids möglichst weit auszudehnen und so viele Komponenten wie möglich unter seine Kontrolle zu bringen. Bei diesem Vorhaben kann der Angreifer sich die Mechanismen des Grid zunutze machen und einfach auf allen erreichbaren Ressourcen einen von ihm speziell präparierten

[8] und ggf. länger, wenn Proxy Renewal möglich ist; s.o.

Grid-Job abgeben. Dieser könnte ein Rootkit enthalten, das auf allen Knoten eines Clusters installiert wird, oder andere Malware auf den angegriffenen Ressourcen starten.

Indem der Angreifer auf jeder erreichbaren Komponente weitere Angriffe versucht, kann er nun sukzessive die gesamte Grid-Infrastruktur kompromittieren. Da er hierbei nur grid-eigene Kommunikationswege nutzt (wie etwa die Abgabe von Jobs mittels des WS-GRAM oder die Kommunikation über einen sicheren Kanal per GSI-SSH), ist eine Detektion über ein herkömmliches IDS praktisch ausgeschlossen.

Die Nutzung bereits kompromittierter Hosts als Sprungbrett für weitere Angriffe ist ein weitverbreitetes Phänomen im Security Engineering. Durch eine Kette kompromittierter Rechner in verschiedenen Organisationen können Angreifer ihre Spuren so weit verwischen, dass eine rechtliche Verfolgung quasi unmöglich wird. Es ist zu erwarten, dass Angreifer auf eine Grid-Infrastruktur ähnlich vorgehen.

3.6 Konsequenzen für den Nutzer

Für Nutzer einer Grid-Infrastruktur ergibt sich nun eine denkbar ungünstige Situation. Ihre Proxy-Credentials können ohne ihr Zutun missbraucht werden und sinnvolle Gegenmaßnahmen können vom Nutzer nicht ergriffen werden. Außerdem wird ein wirksamer Schutz schon allein dadurch verhindert, dass der Nutzer einen typischen Missbrauchsfall erst dann entdecken wird, wenn er bereits geschädigt wurde.

3.6.1 Fehlende Handlungsfreiheit

Das Konzept des Grid Computing sieht vor, dass Nutzer die Rechenleistung des Grid jederzeit abrufen und es dabei als „Black Box“ behandeln können. Obgleich dieser Anspruch in der derzeitigen Praxis nicht erfüllt wird, sind die Vorgänge bei der Authentifizierung und Autorisierung von Nutzern gegenüber Ressourcen nicht transparent. Nutzer sind nicht in der Lage, mögliche Gefährdungen einzuschätzen und können erfolgten Missbrauch weder verfolgen noch eindämmen.

Die fehlende Möglichkeit, den Missbrauch von Proxyzertifikaten einzudämmen, ist im Prinzip der Delegation und des Single Sign-On begründet. In einer auf traditionellen Benutzername/Passwort-Kombinationen basierenden Sicherheitsinfrastruktur genügt es in der Regel, nach Erkennen eines Missbrauchs das oder die betroffenen Passwörter zu ändern, um Missbrauch sofort zu stoppen. In einer PKI muss das

betroffene Zertifikat vom Aussteller zurückgezogen (revoked) werden. Im Falle CA-generierter End-Entity Credentials ist diese Stelle die CA; es muss also ein neues Nutzerzertifikat ausgestellt und das alte eingezogen werden. Proxyzertifikate hingegen können nicht revoked werden, da sie nicht von einer vertrauenswürdigen Stelle direkt ausgestellt wurden. Wie in Abschnitt 2.1.2 erläutert, existiert im Grid derzeit keine Möglichkeit, Proxyzertifikate zurückzuziehen. Die einzige mögliche Maßnahme ist derzeit also die Revokation des EEC – und deren Neuausstellung ist mit teilweise hohem administrativen Aufwand verbunden.

3.6.2 Direkte Schädigung des Nutzers

Durch den Missbrauch seiner Credentials wird der Nutzer direkt geschädigt, wenn in seinem Namen Ressourcen genutzt werden, deren Nutzung eine Autorisierung voraussetzt. Dies gilt insbesondere in dem Fall, in dem eine nutzungsbasierte Abrechnung und Rechnungslegung (Accounting/Billing) stattfindet – der legitime Nutzer muss nun die Kosten für eine Nutzung tragen, die er nicht selber zu verantworten hatte.

Zudem setzt der Verlust der Proxy-Credentials die Daten des Nutzers unkalkulierbaren Gefahren aus, denn sie können vom Angreifer kopiert, aber auch beliebig manipuliert werden. Die Löschung wichtiger Experimentaldaten kann verheerende Folgen für ein auf Grid-Computing basierendes Projekt haben, wenn diese nicht einfach reproduzierbar sind (z.B. bei Experimenten am Large Hadron Collider des CERN). Gegen mißbräuchliche Manipulation von Daten mittels gestohlener Credentials sind auch mehrstufige Backup- und Redundanzkonzepte wie sie am LCG, also der Grid-Infrastruktur des LHC, implementiert werden, machtlos, sofern der Angreifer mit gestohlenen Nutzercredentials Zugriff auf alle Kopien der betroffenen Daten erlangt.

3.6.3 Indirekte Schäden

Der Missbrauch von Proxy-Credentials kann jedoch auch mittelbare Schäden verursachen, die nicht den Nutzer direkt, sondern andere Grid-Teilnehmer oder sogar die Grid-Infrastruktur treffen. So wird der Angreifer in der Regel versuchen, mithilfe der gestohlenen Credentials weitere Nutzerdaten und -Credentials auszuspähen, um diese dann ebenfalls mißbräuchlich zu nutzen. Es ergibt sich somit ein Schneeball-

effekt, der sich durch die gesamte Grid-Infrastruktur fortsetzen kann. Begeht der Angreifer Straftaten (etwa Datenveränderung und das Ausspähen von Daten, die in Deutschland nach §303a und §202a als Straftatbestände gelten), so kann der legitime Nutzer, dessen Zugangsdaten dazu verwendet wurden, zur Rechenschaft gezogen werden.

Bemerkt ein Administrator die missbräuchliche Nutzung rechtzeitig, so wird er bis zur Klärung des Sachverhalts den Grid-Zugang des Nutzers sperren und ggf. dessen Zertifikat einziehen (Revocation). Diese berechtigte und korrekte Reaktion bedeutet jedoch für den legitimen Nutzer eine Nichtverfügbarkeit des für ihn wichtigen Grid-Dienstes und bildet somit aus seiner Perspektive ein „Denial of Service“-Szenario ab.

Da nach gängiger Rechtsauffassung jede virtuelle Organisation in Deutschland wie eine Gesellschaft bürgerlichen Rechts zu behandeln ist, haften alle Mitglieder der VO für das Fehlverhalten des Einzelnen gesamtschuldnerisch. Somit ergeben sich für VO-Mitglieder auch dann unüberschaubare Risiken, wenn sie selber nicht direkt vom Missbrauch eines Proxy-Credentials betroffen sind.

3.7 Fazit

Derzeitige Grid-Infrastrukturen, insbesondere das D-Grid, bieten dem Nutzer keine Möglichkeit, den Missbrauch seiner delegierten Credentials zu detektieren und zu verhindern. Gleichzeitig ergeben sich weitreichende und für Nutzer nicht zu überblickende Risiken aus der Nutzung des Grid mit Proxy-Credentials. Dieses Spannungsfeld zu lösen, ist nach Auffassung des Autors nur möglich, indem entweder die Risiken eingedämmt werden – was jedoch derzeit technisch nicht als machbar angenommen werden kann – oder indem dem Nutzer die Möglichkeit geboten wird, die Verwendung seiner delegierten Rechte so genau wie möglich zu kontrollieren. Ein System, das eine solche Kontrolle ermöglicht, soll im weiteren Verlauf dieser Arbeit vorgestellt werden.

Kapitel 4

Entwurf eines Systems zum Proxy-Auditing

Es soll mittels eines neuen Dienstes im Grid die Möglichkeit geschaffen werden, die Nutzung von Proxy-Credentials in einer Grid-Infrastruktur nachzuverfolgen und somit Mißbrauch zu erkennen. Aus den Gegebenheiten aktueller Grid-Infrastrukturen, aber auch anerkannten Paradigmen und Standards ergeben sich konkrete Anforderungen für ein solches System, auf die im Folgenden eingegangen werden soll.

4.1 Einleitung

Die im vergangenen Kapitel skizzierten Bedrohungsszenarien richten sich gegen alle Elemente, die für Grid-Nutzer von Wert sind. Die Manipulation von Jobs oder die unrechtmäßige Verwendung von Rechenressourcen durch Angreifer kann für legitime Nutzer zu empfindlich hohen Kosten führen und die Manipulation von Daten kann wissenschaftliche Projekte gefährden oder jahrelange Arbeit zunichte machen. Einen wirksamen Schutz gegen den Delegationsmissbrauch gibt es derzeit jedoch nicht. Vorhandene Ansätze zielen darauf ab, das Missbrauchspotential zu verringern, können es aber nicht aus der Welt schaffen.

Ein zentrales Paradigma des Grid sieht vor, dass es sich dem Nutzer gegenüber wie eine „Black Box“ verhält und dieses Verhalten ist meistens unproblematisch. Wenn das Grid ordnungsgemäß funktioniert, benötigen Nutzer keine detaillierten Informationen über Umfang und Verwendung delegierter Rechte. Findet jedoch ein Missbrauch durch Angreifer statt, die unverschlüsselte Proxyzertifikate für ihre eigenen Zwecke missbrauchen, steht Nutzern und Administratoren die zuvor noch nützliche Informationssparsamkeit bei der Analyse und Bekämpfung der Angriffe im Wege. Nutzer, die einen Credentialdiebstahl vermuten, können diesen nicht nachweisen und sind gezwungen, die (bisweilen über eine Woche dauernde) Zeitspanne bis zum natürlichen Ablaufen des mutmaßlich gestohlenen Credentials abzuwarten.

Es ergibt sich also für den Nutzer der Bedarf nach Informationen über die Verwendung seiner „Assets“[1] im Grid. Diese Informationen betreffen alle Aspekte der Grid-Nutzung: Authentifizierung mittels Proxy-Credentials, Weitergabe von Rechten mit Delegationen, Zugriffe auf Ein- und Ausgabedaten, Nutzung von Compute-Ressourcen und Mitgliedschaft in virtuellen Organisationen. Mit den derzeit vom Globus Toolkit zur Verfügung gestellten Werkzeugen kann dieser Bedarf jedoch nicht gedeckt werden.

Die Komponenten des Globus Toolkit halten jeweils eigene Möglichkeiten zur Nachverfolgung verschiedenster Aktivitäten und Vorgänge bereit, die jedoch nicht standardisiert sind.

- Das „GRAM Audit Logging“, dessen Name eine Nähe zum hier vorgestellten Konzept suggeriert, erzeugt Einträge über die Art und Dauer von Grid-Jobs, die als Grundlage für ein Accounting und Billing dienen können, ist jedoch für

[1] also seiner Daten, Job-Beschreibungen, Credentials und Nutzerprivilegien

Sicherheitszwecke nicht hilfreich (siehe Abs. 1.5.7).

- Die Protokollfunktion des Globus WS-GRAM und des Delegation Service meldet zwar die Abgabe eines Grid-Jobs, nicht jedoch die Erzeugung eines neuen Proxy-Credentials und dessen Verwendungszweck.

- Der GridFTP-Server verfügt über eine Möglichkeit, Datentransfers zu protokollieren, jedoch können diese Informationen nicht auf eine sichere Art und Weise nach außen kommuniziert werden.

- Der MyProxy-Dienst kann die Anforderung eines neuen Credentials zwar protokollieren, erhält jedoch über mit diesem Credential gebildete Ableitungen keine Informationen und kann diese demnach auch nicht weitergeben (siehe Abs. 1.5.1).

Protokolle und Loginformationen werden zudem nicht in einem für den Endnutzer verwertbaren Format aufbereitet und sind nur in den seltensten Fällen für ihn direkt einsehbar. Bei Logdateien handelt es sich in aller Regel – so auch beim Globus Toolkit – um interne Dokumente, die zur Fehlersuche und Sicherstellung ordnungsgemäßer Funktion gedacht sind, nicht zur Kontrolle durch die Endnutzer.

Den Komponenten des Globus Toolkit fehlt somit eine einheitliche Schnittstelle und Systematik zur Meldung sicherheitsrelevanter Ereignisse und Aktionen an den Nutzer. Dieser Mangel wird besonders evident, wenn man die Logging-Funktionen des Globus Toolkit in Hinblick auf die Delegation von Proxy-Credentials untersucht. Keine der Komponenten sieht dezidierte Logdatei-Eintragungen für die Erzeugung einer neuen Ableitung vor. Auch die Nutzung von Proxy-Credentials wird, wenn überhaupt, nur in einer sehr grobgranularen Aggregationsstufe festgehalten.

Möchte ein Nutzer, etwa weil er einen Missbrauch seiner Grid-Credentials befürchtet, Zugriff auf die Logdateien aller möglicherweise im Zuge dieses Missbrauchs genutzten Grid-Komponenten erhalten, so wird er sich an jeden Administrator jeder Grid-Ressource wenden müssen, da der Angreifer sich mit entwendeten Proxy-Credentials (wie in Kapitel 3 aufgezeigt) frei im Grid bewegen kann. Der entstehende Arbeits- und Zeitaufwand für Nutzer und Administratoren steht in keinem Verhältnis zum Nutzen. Zudem ist die Zeitkomponente bei Aufklärung und Verhinderung höchst wichtig; je mehr Zeit zwischen dem Identitätsdiebstahl und der Aufdeckung desselben vergeht, desto höher ist das Schadenspotential.

Eine zusätzliche Komponente für das Globus Toolkit sollte also, um einen Sicherheitsgewinn zu erreichen, die Weitergabe sicherheitsrelevanter Informationen an die betroffenen Nutzer automatisch zeitnah ermöglichen, während sich für diejenigen Nutzer, die keine detaillierten Informationen benötigen, keine Änderungen ergeben.

Da – wie oben ausgeführt – eine Auditierung für zahlreiche verschiedene Grid-Assets in Frage kommt, steht für eine erste Ausprägung eines Auditing-Dienstes die Frage im Raum, welche Assets vorrangig auditiert werden sollen, um so viele Anwendungsfälle wie möglich abzudecken.

Es liegt nahe, Grid-Credentials für diese erste Anwendung auszuwählen, denn diese dienen als hauptsächliches Mittel zur Authentifizierung im Grid und erfüllen für ihren Besitzer somit eine zentrale Aufgabe. Ohne gültiges Credential ist über die Schnittstellen des Grid kein Zugriff auf andere Assets möglich. Sie stellen somit dasjenige Asset eines Nutzers dar, dessen Relevanz für das Auditing am höchsten ist.

4.1.1 Notwendigkeit einer zusätzlichen Komponente

Im vorigen Abschnitt wurde gezeigt, dass eine dezidierte Komponente für die Nachverfolgung und Missbrauchserkennung bei Proxy-Credentials im Globus Toolkit derzeit fehlt. Auch vorhandene Komponenten wie etwa das sog. GRAM Audit Logging bieten nicht die hier intendierte Grundfunktionalität. Es ist daher notwendig, eine zusätzliche Komponente für das Globus Toolkit in der für die relevanten nationalen Grid-Infrastrukturen gebräuchlichen Version zu konzipieren und zu implementieren, um das Ziel der lückenlosen Auditierung von Proxy-Credentials im Grid zu erreichen.

4.2 Nicht-funktionale Anforderungen

An das Auditing-System werden – da es sich harmonisch in den Kontext der es umgebenden Grid-Infrastruktur einfügen soll – verschiedene nichtfunktionale Anforderungen gestellt. Diese Anforderungen ergeben sich insbesondere aus den grundsätzlichen Anforderungen an Komponenten eines Grid und aus Paradigmen der verwendeten Middleware, des Globus Toolkit.

Vollständigkeit der Lösung

Ein Auditing-System sollte konzeptionell alle Komponenten einer Grid-Infrastruktur gleichermaßen abdecken können, die mit Nutzer-Credentials in Berührung kommen. Ist diese Anforderung nicht erfüllt, kommt es zu Lücken im Auditing, wenn eine nicht auditfähige Komponente angesprochen wird. Die resultierenden Audit Trails sind in diesem Fall nur noch von begrenzter Aussagekraft.

Zu auditieren sind alle Grid-Komponenten, die eine Authentifizierung mittels Proxy-Credentials unterstützen, mithin also in einer Globus-Infrastruktur alle vorhandenen Komponenten. Für jede Komponente kann eine eigene Auditing-Schnittstelle entwickelt werden. Es stellt sich aber als sinnvoller dar, eine gemeinsame Schnittstelle zu implementieren, die alle Komponenten einer Globus-basierten Infrastruktur einschließt.

Kompatibilität und Interoperabilität

Es ist für den reibungslosen Betrieb einer Grid-Infrastruktur unablässig, dass der Einsatz einer neuen Komponente nicht zu unerwünschten Wechselwirkungen führt, die andere Teile des Grid in ihrem Betrieb beeinträchtigen. Daher ist bei Entwurf und Umsetzung des Auditing-Systems zu beachten, dass die Kompatibilität mit dem Globus Toolkit voll gewährleistet ist. Desweiteren darf in einer Grid-Umgebung, die mehrere Middlewares unterstützt (D-Grid, Nordugrid), die zu entwickelnde Lösung keine Probleme im Betrieb der weiteren Middlewares auslösen.

Die Kompatibilität mit den im Grid verwendeten Protokollen und Standards für Verschlüsselung, Authentifizierung und Zertifizierung (X.509 [ITU05], RFC 3820 [TWE+04] etc.) muss ebenfalls sichergestellt werden, da eine reibungslose Integration in die Grid-Umgebung sonst ebenso wenig gewährleistet werden kann wie ein zuverlässiges Auditing. Das Grid funktioniert nur durch die Definition und strikte Einhaltung von Standards für alle Bereiche, in denen Komponenten miteinander Daten austauschen. Ian Foster stellt die Einhaltung von Standards in seinem Positionspapier „What is the Grid?“ ([Fos02], siehe auch Abschnitt 2.2) als eine der drei wichtigsten Kennzeichen eines Grid heraus.

Wie Foster weiter ausführt, hält er die Verbindung einzelner Grid-Infrastrukturen miteinander für eines der am vordringlichsten zu lösenden Probleme. Obwohl diese Aussage bereits im Jahr 2002 getätigt wurde, ist sie noch immer gültig. Standards,

wie sie vom Open Grid Forum (OGF) oder der Open Grid Services Architecture (OGSA) definiert und durchgesetzt werden, erleichtern die Kommunikation innerhalb des weltweiten Grid und Entwicklung neuer Dienste und Protokolle erheblich.

Ein Auditing-System muss, auch im Sinne der im vorangegangenen Abschnitt geforderten Interoperabilität, diese Standards bei der Interaktion mit anderen Grid-Komponenten, aber auch in der Kommunikation seiner eigenen Komponenten, beachten.

Sicherheit

Ein besonderer Schwerpunkt in der Anforderungserhebung für das vorliegende Projekt sind die Sicherheitsanforderungen – Schließlich soll durch das Auditing von Proxy-Credentials das Sicherheitsniveau im Grid erhöht werden und mangelnde Abdeckung grundsätzlicher Sicherheitskriterien wäre hier kontraproduktiv.

Nichtfunktionale Anforderungen bestehen sowohl im Unterbereich der Security, als auch der Safety. Wie bereits in den einführenden Anmerkungen zu Security-Aspekten im Grid in Abschnitt 2.2.3 erläutert, bestehen die wesentlichen nichtfunktionalen Sicherheitsanforderungen an eine Grid-Komponente im Wesentlichen aus dem folgenden drei Anforderungen: Nichtbestreitbarkeit, Authentizität und Vertraulichkeit. Im Bereich der Betriebssicherheit (also der Safety) ist im Wesentlichen die Verfügbarkeit, also die zuverlässige Erreichbarkeit und Funktion, ein wichtiges nichtfunktionales Kriterium.

Non-Repudiation

Wurde ein Proxy-Credential von einem Angreifer übernommen und genutzt, um in einer auditingfähigen Grid-Infrastruktur missbräuchliche Handlungen vorzunehmen, so werden diese Handlungen durch Auditing-Datensätze dokumentiert und somit dem wahren Besitzer des Proxy-Credentials verdeutlicht. Nur, wenn eine Non-Repudiation (Nichtabstreitbarkeit) der vorliegenden Audit-Records gegeben ist, können die Auditing-Informationen zweifelsfrei als verwertbare Indikatoren für Missbrauch dienen.

Vertraulichkeit

Kommunikation im Grid unterliegt besonderen Anforderungen an die Vertraulichkeit. Diese erstrecken sich etwa auf die Übertragung von Ein- und Ausgabedaten, aber auch auf die Abgabe von Grid-Jobs. Ein Audit Record ist ein Datum mit direktem Personenbezug (nämlich zum Inhaber des auditierten Proxy-Credentials), daher ist stets sicherzustellen, dass die Auditing-Daten vertraulich übertragen und vom Empfänger – dem Auditing-System – ebenso behandelt werden.

Die Übertragung von Auditingdaten über das Netzwerk von Dritten darf also nicht abhörbar sein, das Auditingsystem muß zudem über geeignete Maßnahmen zur Zugriffskontrolle sicherstellen, daß personenbezogene Daten nur dem betroffenen Nutzer zugänglich sind.

Authentizität

Zunächst ist sicherzustellen, dass alle an das Auditing-System übertragenen Daten, also mithin die Audit Records, authentisch sind. Wären Audit Records beliebig von Dritten fälschbar, so könnte ein Angreifer über massenhafte Generierung falscher Daten das System, insbesondere seine Erkennungsrate tatsächlichen Missbrauchs, massiv beeinflussen. Desweiteren muss auch die Authentizität aller Kommunikation mit dem Auditing-System gewährleistet sein, denn sonst wären Auditing-Daten und die vom System generierten Meldungen nicht eindeutig zuzuordnen, was das System nutzlos machen würde.

Verfügbarkeit

Ein Auditing-System stellt eine wichtige Ergänzung zur Sicherheitsinfrastruktur des Grid dar. Daher ist es notwendig, dass der Dienst allen Nutzern verlässlich zur Verfügung steht. Diese Verlässlichkeit schließt die Zugreifbarkeit unter Normalbedingungen ein, umfasst aber insbesondere auch, dass der Dienst auch bei Extrembedingungen nutzbar ist. Angreifern darf es nicht möglich sein, das Auditing-System mit massenhaftem Versand von (authentischen oder gefälschten) Daten zu überlasten; auch eine Hochlastsituation, die durch massenhafte legitime Nutzung des Auditing-Dienstes eintritt, muss das Auditing-System verkraften.

4.2.1 Skalierbarkeit

Die im vorigen Unterabschnitt geforderte Verfügbarkeit sicherzustellen, wird in einer hochproduktiven Grid-Umgebung nur möglich sein, indem mehrere Auditing-Dienste parallel betrieben werden. Diese Skalierbarkeit muss beim Entwurf des Dienstes berücksichtigt werden, so dass etwa ein Auditing-Dienst pro Grid-Site oder pro virtueller Organisation betrieben werden kann, ohne andere gleichartige Dienste negativ zu beeinflussen. Auch eine Skalierbarkeit über die Grenzen einzelner nationaler Grids hinweg in einen internationalen Rahmen ist ohne weitere Anpassungen möglich zu machen.

4.3 Funktionale Anforderungen

Aus den im vergangenen Kapitel 3 erarbeiteten Bedrohungsszenarien und dem ebenda benannten Bedarf für ein Auditing-System ergeben sich verschiedene funktionale Anforderungen, die das System zum Proxy-Auditing erfüllen muss.

Erstellung von Auditing-Einträgen

Für jede Aktion, die unter Vorlage eines auditierten Proxy-Credentials ausgeführt wird, soll eine Meldung („Audit Record“) an das Auditing-System ergehen. Eine solche Meldung soll alle Daten enthalten, die notwendig sind, um sie mit anderen Auditingdaten zu verknüpfen und gegebenenfalls einen Missbrauch zu erkennen. Die Möglichkeit, solche Daten während einer GSI-gestützten Kommunikation abzugreifen und an einen Auditing-Dienst zu versenden, ist jedoch in keiner Komponente des Globus Toolkit vorgesehen. Es ist daher notwendig, jede Komponente, also etwa den Globus GRAM, den GridFTP-Server oder auch MyProxy, auf Anwendungsebene zu modifizieren, um Auditing zu ermöglichen. Eine weitere Möglichkeit, Auditingdaten zu erheben, ist die Modifikation der zentralen GSI-Bibliotheken.

Sichere Übertragung

Einmal erhobene Auditingdaten dürfen von einem Angreifer weder eingesehen noch modifiziert werden können; auch soll ein nachträgliches Abstreiten auditierter Handlungen nicht möglich sein. Es sind daher an die Übertragung von Auditingdaten die

in Abschnitt 4.2 aufgeführten Maßstäbe anzulegen: Authentizität, Nichtabstreitbarkeit und Vertraulichkeit der Auditingdaten sind zu gewährleisten.

Verarbeitung der Auditingdaten

Sind Audit Records vom Auditingdienst empfangen und ihre Authentizität festgestellt worden, so müssen diese Daten zunächst gespeichert werden, bevor sie weiterverarbeitet werden (s.u., Missbrauchserkennung). Diese Speicherung soll nach möglichst offenen Standards und unter Beachtung von Datenschutz und Sicherheit erfolgen. Gleichzeitig sollen die Daten für nachgelagerte Komponenten des Auditing-Systems leicht und schnell abrufbar sein.

Es empfiehlt sich daher, die Auditingdaten in einer relationalen Datenbank abzuspeichern, die mittels strukturierter Abfragesprache (SQL) bedient werden kann.

Missbrauchserkennung

Die nicht aggregierten Rohdaten des Auditing sind für den Nutzer, aber auch für Administratoren nur schwer zu durchschauen. Insbesondere die manuelle Ermittlung missbräuchlicher Grid-Nutzung ist höchst schwierig. Daher soll das Auditing-System diese Aufgabe automatisiert durchführen, um dem Nutzer eine qualifizierte Einschätzung zu vermitteln, ob Credentialmißbrauch vorliegt. Die Erkennungskomponente soll die vielfältigen (legitimen) Nutzungsmöglichkeiten des Grid in Betracht ziehen und eine möglichst geringe Rate an Falschmeldungen (False Positives/Negatives) erreichen. Dem Nutzer soll eine leicht verständliche und verläßliche Angabe zum Status seiner Grid-Credentials gemacht werden..

Nutzerzentrische Optionen

Dem Nutzer soll die Entscheidung, ob die Nutzung seiner Proxy-Credentials auditiert werden soll, selbst überlassen bleiben. Es ist daher vorzusehen, dass das Auditing von Proxy-Credentials für jedes individuelle Credential aktiviert oder deaktiviert werden kann. Diese Entscheidung ist den auditingfähigen Grid-Komponenten auf eine geeignete (also nach Abs. 4.2 sichere) Art mitzuteilen; diese müssen sich dann daran halten.

Es ist zu vermeiden, dass die Entscheidung des Nutzers aufgrund technischer Fehler vom Auditing-System nicht berücksichtigt wird; im Zweifelsfall soll das Sys-

tem ein Vorgehen verfolgen, das minimale Sicherheitsimplikationen mit sich bringt. Zudem sollte der Nutzer die Möglichkeit haben, zu bestimmen, wie mit nicht auditingfähigen Ressourcen umgegangen werden soll.

Benutzbarkeit

Damit der Endnutzer, dessen Credentials auditiert werden, adäquat mit dem Auditing-System interagieren kann, ist eine Benutzeroberfläche vorzusehen. Diese besteht sinnvollerweise aus zwei Teilen: Zunächst aus einer kommandozeilenbasierten Schnittstelle, mithilfe derer der Nutzer das Auditing von Proxy-Credentials aktiviert oder deaktiviert, aber auch aus einer Benutzeroberfläche zur Einsicht der aufbereiteten Auditing-Daten.

Hier ist es zweckmäßig, eine webbasierte Oberfläche zu schaffen, mittels derer der Nutzer die über ihn erhobenen Daten einsehen kann und die ihm ebenfalls eine Möglichkeit bietet, die Ergebnisse der Missbrauchserkennung zu überblicken. Zugriffe auf diese webbasierte Oberfläche sollten zudem nur unter Berücksichtigung der in Abschnitt 4.2 aufgeführten Sicherheitskriterien erfolgen.

4.4 Entwurf

Wie im vorigen Kapitel erläutert, sollte eine standardkonforme, sichere Lösung für das Auditing von Grid-Proxy-Credentials gefunden werden, die zu bestehenden Komponenten so weit wie möglich kompatibel ist und sich in die vorhandene Grid-Landschaft problemlos einfügt. Diese Anforderungen dienten als Grundlage für die Entwicklung des Auditing-Systems, dessen Architektur im Folgenden zunächst skizziert und dann detailliert beschrieben werden soll.

Die grundsätzliche Architektur des Auditing-Systems ergibt sich mehr oder minder direkt aus den in Abschnitt 4.3 genannten funktionalen Anforderungen. Benötigt wird zunächst eine Möglichkeit, Auditing-Daten bei der Grid-Nutzung zu erheben. Diese müssen dann an einen geeigneten Dienst übermittelt werden, wo sie in einer Datenbank gespeichert werden. Die Auswertung der gesammelten Daten geschieht mit einem System zur automatisierten Missbrauchserkennung; der Nutzer kann mit dem gesamten System über eine webbasierte Schnittstelle kommunizieren.

Für jede Teilkomponente existierten verschiedene mögliche Implementierungsansätze, die in den folgenden Abschnitten evaluiert werden.

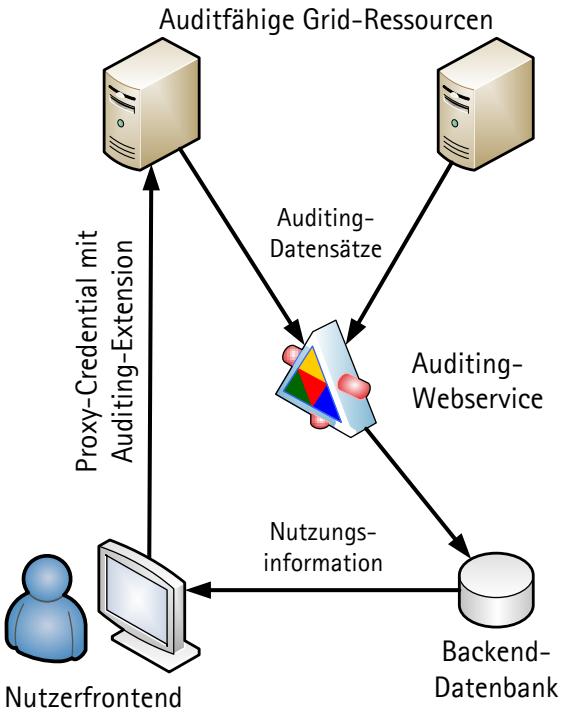


Abbildung 4.1: Grundsätzlicher Aufbau des Auditing-Systems

4.4.1 Auditing auf Grid-Komponenten

Zunächst war zu untersuchen, wie das Auditing der Zertifikatsnutzung bei vorhandenen Komponenten des Globus Toolkit ermöglicht werden kann. Um zu erreichen, dass jede Zertifikatsnutzung auditiert wird, sind grundsätzlich zwei Ansätze möglich: Der erste Ansatz sieht eine spezialisierte Modifikation jeder einzelnen Komponente auf Anwendungsebene vor, während ein zweiter möglicher Ansatz die Sicherheitsbibliotheken des Globus Toolkit modifiziert, um dasselbe Ziel zu erreichen. Beide Ansätze sollen zunächst kurz diskutiert werden.

Modifikation auf Anwendungsebene

Das Globus Toolkit enthält einige aktive Komponenten, die im Einsatz mit Proxy-Credentials in Berührung kommen. Zu diesen Komponenten zählen unter Anderem:

- Nutzer-Werkzeuge zur Verwaltung von Proxy-Credentials (`grid-proxy-init`, MyProxy, GRAM Delegation Service) und zur Job Submission (`globusrun`, `globusrun-ws`)
- Jobverwaltung und -bearbeitung (GRAM, WS-GRAM)

- Datendienste (GridFTP, dCache, OGSA-DAI) und Fernverwaltung (GSI-SSH)

Jeder dieser Dienste verarbeitet Proxy-Credentials und muss daher für das Auditing berücksichtigt werden. Damit ist eine Modifikation von etwa einem Dutzend verschiedener Anwendungen zu implementieren, um das Auditing an jeder Stelle zu ermöglichen.

Die für das Auditing auf Anwendungsebene notwendigen Änderungen in jeder dieser Applikationen separat einzubauen, stellt zunächst einen hohen Implementationsaufwand dar, der zudem die Fehlerwahrscheinlichkeit deutlich erhöht. Diese Überlegung allein ist jedoch nicht ausreichend, um diesen Ansatz zu verwerfen; zwei weitere wichtige Argumente sprechen gegen ihn.

Zum Einen ist die Akzeptanz durch das Globus Consortium beim Globus Toolkit ein wichtiger zu berücksichtigender Faktor. Eine Auditing-Infrastruktur, für die Modifikationen an allen essentiellen Globus-Komponenten notwendig werden, würde in der Entwicklergemeinschaft schwerlich auf viel Akzeptanz stoßen. Schließlich ist die Gefahr groß, dass Fehler in einzelnen Teilimplementationen das Gesamtsystem negativ beeinflussen. Zum Anderen stehen die für das Auditing notwendigen Informationen auf Anwendungsebene unter Umständen nicht oder nicht mehr zur Verfügung. Die GSI kapselt kryptographische Vorgänge wie die gegenseitige Authentisierung gegenüber der Anwendungsebene. So stehen einige für das Auditing notwendige Daten der Anwendungsebene womöglich nicht mehr zur Verfügung. Dazu zählen insbesondere die Willensäußerung des Nutzers zum Auditing (siehe hierzu auch Abschnitt 4.4.2) sowie weitere Metadaten seiner Proxyzertifikate. Jene Informationen sind nur auf der GSI-Ebene verfügbar. Daher erscheint es zunächst nicht wünschenswert, die Auditierung auf Anwendungsebene durchzuführen.

Auditing auf GSI-Ebene

Wie in Abschnitt 2.2.3 ausgeführt und in Abb. 4.2 illustriert, nutzt jede Grid-Komponente die Sicherheitsarchitektur GSI, um Vertraulichkeit, Authentizität und Nichtabstreitbarkeit der Kommunikation im Grid zu garantieren. Somit muß jede Komponente programmatisch auf eine Implementation der GSI zurückgreifen, um die notwendige Basisfunktionalität zu implementieren. Die GSI erscheint demnach als logischer Ansatzpunkt, um die für das Auditing notwendigen Modifikationen in die relevanten Globus-Komponenten einzubringen.

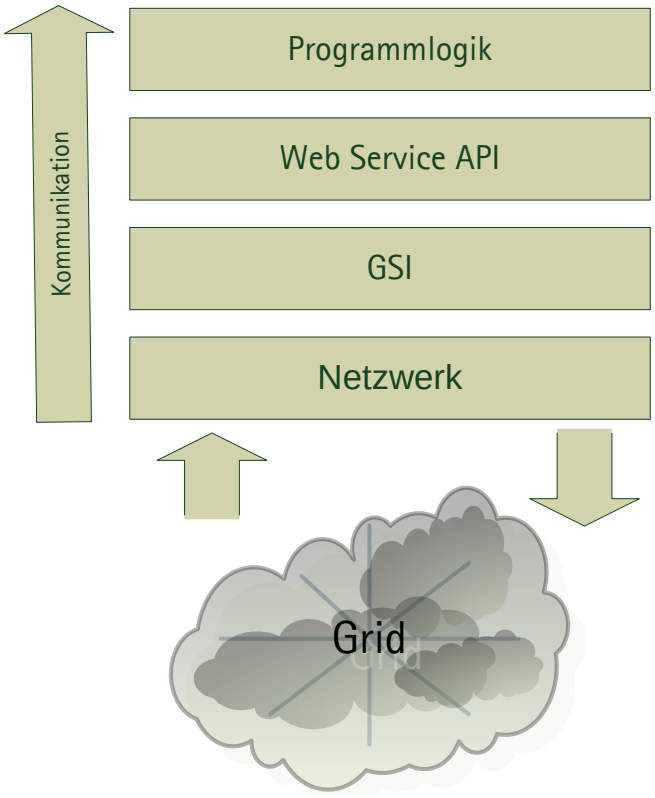


Abbildung 4.2: Schema der Protokollebenen im Globus Toolkit

Komponenten des Globus Toolkit sind derzeit in zwei verschiedenen Programmiersprachen implementiert: Java und C. Während C für alle Komponenten genutzt wird, die eigene, nicht webservice-gestützte Kommunikationsprotokolle verwenden (GridFTP, GSI-SSH, preWS-GRAM, MyProxy, dCache), kommt bei allen auf Webservices aufbauenden Komponenten, insbesondere also beim WS-GRAM oder OGSA-DAI, Java zum Einsatz. Für beide Sprachen existieren GSI-Implementationen in den entsprechenden Basisbibliotheken. Gegenüber der Modifikation auf Anwendungsebene ergibt sich somit ein deutlicher Effizienzgewinn, denn anstelle von bis zu zwölf unterschiedlichen programmatischen Abbildungen des Auditing sind nun lediglich zwei – je eine Implementation für die Java- und C-GSI-Bibliotheken – zu leisten. Eine fehlerhafte Implementation dieser Änderungen in der GSI gefährdet den sicheren Betrieb der betroffenen Grid-Komponente, durch die Beschränkung auf nur zwei Implementierungsansätze ist jedoch sichergestellt, dass jeweils ausreichende Testmöglichkeiten bestehen.

Desweiteren ist sichergestellt, dass auf der GSI-Protokollebene alle notwendigen Informationen zur Verfügung stehen, um ein zielführendes Auditing durchführen zu können. Das für das Auditing wichtige Proxy-Zertifikat nebst aller Ableitungen steht der GSI-Bibliothek zur weiteren Verarbeitung zur Verfügung und kann somit auch von einer Auditing-Komponente – sofern sie auf GSI-Ebene implementiert wird – verwendet werden.

Ein Nachteil der Implementierung des Auditings auf GSI-Ebene ist jedoch, dass der genaue Verwendungszweck, also etwa der Grund der Authentifizierung, die genauen abgerufenen Dateinamen eines GridFTP-Servers oder auch die Beschreibung eines Grid-Jobs, nicht durch die Auditing-Komponente eingesehen werden kann, da diese Informationen nur einer höheren Protokollebene im oben illustrierten Stack zur Verfügung stehen.

Fazit

Für den intendierten Zweck des vorliegenden Projektes – eine möglichst umfassende Möglichkeit des Auditing von Grid-Proxy-Credentials – überwogen die Vorteile der GSI-Modifikation deutlich die relativ überschaubaren Nachteile. Da im vorliegenden Projekt die Überlegung im Vordergrund stand, die Nutzung von Credentials zunächst anwendungsunabhängig zu auditieren, wurde daher zunächst auf eine GSI-basierte Implementierung zurückgegriffen. Es ist jedoch ohne Weiteres möglich, in

Zukunft anhand der in dieser Dissertation vorgestellten Auditing-Infrastruktur die Nutzung von Proxy-Credentials auch auf Anwendungsebene zu auditieren.

4.4.2 Modifikation der Grid-Proxy-Zertifikate

Wie in den Anforderungen in 4.2 festgestellt, muss eine Erweiterung der Globus-Middleware sich nahtlos in die bestehende Umgebung einfügen, darf also mithin keine unerwünschten Nebeneffekte bei anderen Komponenten erzeugen. Gleichzeitig muß die Notwendigkeit, ein Proxy-Credential zu auditieren, jedoch auch an alle Ressourcen kommuniziert werden. Es ist nicht erwünscht, für diese Kommunikation ein nicht standardisiertes Out-of-Band-Protokoll zu verwenden; die zum Auslösen des Auditings notwendige Kommunikation soll keinen zusätzlichen Overhead erzeugen.

Gleichzeitig ist ein wichtiges in Abschnitt 4.2 genanntes Kriterium, dass die Auditingnachrichten, aber auch die Information, dass ein bestimmtes Credential auditiert werden soll, auf eine sichere Art und Weise übermittelt werden. Das bedeutet insbesondere, dass diese Information nicht nachträglich modifiziert, im Regelfall also entfernt werden kann, um das Auditing auszuhebeln.

Wie in Abschnitt 2.1.4 erläutert, enthält eine aktuelle Grid-Infrastruktur kein (logisch) zentrales Element, das Authentifizierungs- und Autorisierungsinformationen mit allen anderen Elementen austauscht. Es existiert daher keine Möglichkeit, die Nutzung von Credentials an einer zentralen Stelle zu erfassen und ggf. zu speichern. Eine Speicherung ist jedoch notwendig, um die Auditingdaten zu aggregieren. Ein Informationsgewinn ergibt sich schließlich erst, sobald nicht die einzelne Zertifikatsnutzung, sondern der Weg einer Zertifikatskette über deren gesamte Nutzungsdauer betrachtet wird.

Proxy-Credentials werden von ihrem Besitzer mit einer nachprüfbaren digitalen Signatur versehen, die die enthaltenen Metainformationen validiert. Es ist naheliegend, diesen Signaturmechanismus zu benutzen, um die für das Auditing notwendigen Informationen direkt in das Proxy-Zertifikat einzubetten.

Die Architektur der GSI setzt für jegliche Kommunikation eine Authentifizierung der Kommunikationspartner – also in der Regel eines Nutzers und einer Ressource – voraus. Damit stehen im Kontext jeder Grid-Kommunikation genügend Informationen zur Verfügung, um einen Nutzer zweifelsfrei zu identifizieren, auch wenn dieser seine Berechtigungen in Form einer Delegation an eine Ressource abgegeben hat. Zudem stehen das End-Entity-Zertifikat, aber auch alle „Eltern-Delegationen“ (siehe

Abschnitt 2.2.4) zur Verfügung, jedoch jeweils ohne ihre privaten Schlüssel. Somit sind die Eltern-Delegationen zwar nicht für weitere Authentifizierungsvorgänge geeignet, ihren Metadaten können jedoch dank der überprüfbaren digitalen Signatur der Zertifikate vertraut werden.

Es war also eine Möglichkeit zu finden, ein zu auditierendes Proxy-Credential so zu modifizieren, dass auditingfähige Grid-Komponenten eindeutig und nachvollziehbar feststellen können, ob ein Auditing für das betreffende Credential notwendig ist, während nicht auditingfähige Komponenten dieselbe Information ohne Nebeneffekte ignorieren können.

Eine ähnliche Anforderung wurde durch Piger in [Pig08] erfüllt, indem die dort zur nutzerbasierten Restriktion eingesetzten Policies in die digitalen Zertifikate integriert wurden. Die Implementierung erfolgte mit standardisierten Zertifikatserweiterungen. Dieses Vorgehen empfiehlt sich auch als Lösung für die vorliegende Problemstellung. In den folgenden Abschnitten soll untersucht werden, inwiefern sich der Lösungsvorschlag mit den in den Abschnitten 4.2 und 4.3 aufgestellten Anforderungen deckt und welche Vorteile sich ergeben.

X.509-Zertifikatserweiterungen

Der Standard X.509 [ITU05] der ITU-T definiert ein Format für digitale Zertifikate, das in Grid- und Cloud-Infrastrukturen, aber auch in der digitalen Kommunikation per E-Mail und im World Wide Web breite Anwendung findet. Neben den in jedem Zertifikat notwendigen Informationen wie einer eindeutigen Seriennummer, digitalen Signatur und dem Subjekt (also Inhaber) des Zertifikats ist auch eine Möglichkeit vorgesehen, weitere Informationen mittels Zertifikatserweiterungen („Extensions“) einzubringen. Diese Erweiterungen werden anhand einer eindeutigen Kennung (Object Identifier, siehe [ITU08d]) referenziert und im ASN.1-Format [ITU08c] formatiert. Details zur Implementierung einer X.509-Zertifikatserweiterung sind in Abs. 5.4.1 zu finden.

Die Verarbeitung von Zertifikatserweiterungen ist in jeder aktuellen SSL-Implementierung, also insbesondere in den für die GSI relevanten Implementierungen in C und Java bereits enthalten, da auch verschiedene Metainformationen zu digitalen Zertifikaten in Erweiterungen abgelegt werden. Da weder die maximale Größe einer Erweiterung noch ihre Inhalte reglementiert sind (lediglich die Formatierung innerhalb des Zertifikats ist durch ASN.1 festgelegt), können in Zertifikatserwei-

terungen quasi beliebige Daten transportiert werden. Die Spezifikation sieht vor, dass Zertifikatserweiterungen mit einer entsprechenden Variable als **kritisch** oder **nicht-kritisch** markiert werden können. Diese Einteilung hat Auswirkungen auf die Verarbeitung des Zertifikats. In der Spezifikation X.509 wird folgendes Verhalten vorgeschrieben:

> When an implementation processing a certificate does not recognize an extension, if the criticality flag is FALSE, it may ignore that extension. If the criticality flag is TRUE, unrecognized extensions shall cause the structure to be considered invalid, i.e. in a certificate, an unrecognized critical extension would cause validation of a signature using that certificate to fail. When a certificate-using implementation recognizes and is able to process an extension, then the certificate-using implementation shall process the extension regardless of the value of the criticality flag. (X.509, Abs. 7, S. 24)

Um die geforderte maximale Kompatibilität und Interoperabilität mit bestehenden Systemen, die die Auditing-Erweiterung in X.509 Proxy-Zertifikaten nicht erkennen und verarbeiten können, herzustellen, sollte demnach die Erweiterung als nicht kritisch ausgewiesen werden. Nicht auditingfähige Ressourcen können die für das Auditing verwendete Zertifikatserweiterung demnach übergehen, ohne dass das Zertifikat mangels gültiger Signatur unbrauchbar wird.

Die Signatur eines Zertifikats (auch die eines Proxy-Zertifikats) umfasst sämtliche in diesem Zertifikat enthaltenen Daten, auch Zertifikatserweiterungen. Somit ist den Anforderungen an Authentizität und Nichtabstreitbarkeit der in einer X.509-Extension enthaltenen Informationen Genüge getan. Ein Angreifer kann nach aktuellem Stand der Technik ein bereits signiertes digitales Zertifikat nicht nachträglich modifizieren, ohne die Signatur zu zerstören (vgl. Abs. 2.1.1 und 2.1.2).

Da alle verwertbaren Informationen für das Auditing, insbesondere Inhaber-, Gültigkeitsdaten und eindeutige Seriennummern bereits in den Metadaten des Zertifikats enthalten sind, stellt sich die Frage, welche Nutzdaten über die binäre Entscheidung für oder gegen Auditing hinaus mittels einer Zertifikatserweiterung transportiert werden müssen. Hier bietet sich an, dem das Zertifikat auditierenden System mitzuteilen, an welchen Endpunkt die Auditinginformationen mittels eines gesicherten Kommunikationsprotokolls übermittelt werden sollen. Da (wie i.F. in Abs. 4.4.4 erläutert) ein Webservice als Empfänger der Auditingdaten zum Einsatz kommt,

bietet sich die Aufnahme einer URL in die Zertifikatserweiterung an. So verfügt die auditierende Ressource über eine von Angreifern nicht fälschbare Adresse, die Auditingdaten entgegennimmt; ein Unterdrücken der Informationen durch Dritte wird somit zumindest erschwert.

Nutzerzentrische Entscheidung für Auditing

Das Auditing von Grid-Credentials soll in erster Linie die Sicherheit für den Grid-Nutzer erhöhen. Eine Auditierung von Credentials verpflichtend für alle Nutzer einzuführen, könnte jedoch auf datenschutzrechtliche Bedenken stoßen. Es ist daher im Entwurf der Auditing-Systeme vorzusehen, dass die Entscheidung für oder gegen die Auditierung von Proxy-Credentials dem Nutzer überlassen ist und er sie bei jedem neu von ihm delegierten Proxy der ersten Ableitungsebene aufs Neue treffen kann.

Beachtung der Zertifikatskette

Wie in Abs. 2.2.4 erläutert, können von Proxy-Credentials weitere Ableitungen gebildet werden, die ebenfalls zur Authentifizierung und zur Erstellung weiterer Delegationen genutzt werden können. Die Zertifikatserweiterung für das Auditing könnte auch in diesen Ableitungen der zweiten, dritten oder tieferer Ebenen enthalten sein. Ist sie das, ist ihre Authentizität jedoch nicht mehr gewährleistet, da prinzipgemäß jede Grid-Ressource (oder jeder Nutzer, der im Besitz eines Proxy-Credentials ist) weitere Ableitungen signieren und entsprechend mit beliebigen Erweiterungen versehen könnte.

Die Authentizität der Zertifikatserweiterung zum Auditing kann nur dann gewährleistet werden, wenn sie in dem Proxy-Credential enthalten ist, das der Grid-Nutzer mit dem privaten Schlüssel seines EEC signiert. Nur diese Signatur ist garantiert durch den Nutzer persönlich ausgeführt worden (da nur er im Besitz seines privaten Schlüssels ist).

Es ist also bei der Implementierung zu beachten, dass der Nutzerwunsch zum Auditing lediglich in der ersten Delegationsebene durch verarbeitende Ressourcen Beachtung finden darf; eventuelle anderslautende Informationen in weiteren Ableitungen derselben Zertifikatskette sind zu ignorieren.

Fazit

Die notwendigen Modifikationen an Grid-Proxy-Credentials sind technisch ohne Weiteres umsetzbar und lösen verschiedene Probleme. So ist die notwendige Willensäußerung des Nutzers dadurch gegeben, dass dieser eine standardkonforme Zertifikatserweiterung in das Proxy-Credential integriert, das mittels seines EEC signiert wird. Die Nutzdaten dieser Zertifikatserweiterung dienen gleichzeitig als Information über den vom Nutzer intendierten Empfänger der Auditing-Nachrichten. Somit wird durch die Modifikation der Grid-Proxy-Credentials eine authentische und nicht von Angreifern manipulierbare Möglichkeit geschaffen, das Auditing durch Grid-Ressourcen auszulösen.

4.4.3 Modifikation der GSI-Komponenten

Zwar wurde mit der im vorangegangenen Abschnitt erläuterten Modifikation der Grid-Proxy-Zertifikate die Möglichkeit geschaffen, den Nutzerwunsch nach Auditing an Grid-Komponenten zu kommunizieren, diese sind ohne Modifikation jedoch nicht in der Lage, diesem Wunsch zu entsprechen. Wie in Abs. 4.4.1 subsummiert, soll diese Modifikation auf der GSI-Bibliotheksebene stattfinden.

Im Globus Toolkit existieren zwei Implementationen der GSI nebeneinander:

1. Eine Implementation in C, die auf OpenSSL basiert und auf der Dienste wie GridFTP und GSI-SSH aufbauen.

2. Die auf BouncyCastle aufbauende Java-Implementation, die bei allen webservicebasierten Komponenten zum Einsatz kommt.

Im Folgenden (sowie in der in Kapitel 5 beschriebenen beispielhaften Implementierung) soll zunächst auf die Java- und webservicebasierten Komponenten und Bibliotheken eingegangen werden. Einige zentrale Funktionen des Auditing werden durch Modifikationen in diesen Bibliotheken abgedeckt.

Analyse der Zertifikatserweiterung

Jeder Kommunikation im Grid geht der Aufbau eines sicheren Kanals und die gegenseitige Authentifizierung (siehe 2.1.3) voraus. Während dieses Vorgangs tauschen

der Client (also der Nutzer des Proxy-Zertifikats) und der Server (also die zu modifizierende Grid-Komponente) ihre Zertifikate aus; der Client verwendet hier ein Proxy-Zertifikat.

Im Zuge der Prüfung des Proxy-Zertifikats werden dessen Signatur, Gültigkeit und Signaturpfad geprüft. Zusätzlich wird in der für das Auditing modifizierten Version der GSI die Zertifikatserweiterung ausgelesen und analysiert, die den Auditingwunsch des Nutzers enthält. Die Erweiterung wird zunächst auf Existenz untersucht; ist sie im Proxy-Zertifikat enthalten, bedeutet das für das Auditing-Subsystem, dass für die Kommunikation ein Auditing-Track zu erstellen ist. Aus den Nutzdaten der Erweiterung wird dann eine Liste von einer oder mehreren URLs extrahiert, die als Ziel(e) für den später erfolgenden Versand der Auditing-Tracks dienen.

Zusammenstellung von Auditing-Einträgen

Nachdem die modifizierten GSI-Bibliotheken über die Anweisung des Nutzers, eine Auditierung der gerade stattfindenden Kommunikation durchzuführen, informiert sind, müssen die entsprechenden Informationen gesammelt und für den Versand an die empfangende Stelle – den Auditing-Webservice – vorbereitet werden.

Wie in Abs. 4.3 erläutert, müssen verschiedene Daten über die Zertifikatsnutzung aggregiert und aufbereitet werden. Diese Daten sind insbesondere:

- Quell- und Zielhost der auditierten Kommunikation
- Datum und Uhrzeit der auditierten Aktion
- Art der auditierten Aktion (Authentifizierung oder Delegation)
- Subjekt des Auditing, also Name des Zertifikatsinhabers
- Seriennummer des auditierten Zertifikats

Die Informationen sollen nach Möglichkeit aus einer nachvollziehbaren und nicht von Angreifern fälschbaren Quelle stammen. Während der Zielhost der auditierten Kommunikation leicht zu ermitteln ist (es handelt sich um den Kommunikationspartner, der das Auditing durchführt), kann der Quellhost aus den durch den Netzwerkstack zur Verfügung gestellten Protokollinformationen zuverlässig ermittelt werden. Der Auditingdienst kann zusätzlich über die DNS-Einträge die Plausibilität dieser Daten prüfen. Datum und Uhrzeit ergeben sich aus der lokalen Systemzeit auf dem

auditierenden Host; auch die Art der auditierten Aktion steht auf GSI-Ebene zur Verfügung (s.u. 5.2). Das Subjekt des Auditing, also der „Distinguished Name“ des Proxy-Zertifikats, ist ebenso wie dessen Seriennummer Bestandteil der signierten Zertifikatsinformationen und wird vom GSI-Subsystem nach erfolgter gegenseitiger Authentisierung zur Verfügung gestellt.

Sicherer Versand von Auditing-Einträgen

Der Versand von Auditing-Informationen erfolgt über eine standardisierte Web-Service-Schnittstelle (WSRF [CFF+04]). Vor dem Versand müssen die zuvor erhobenen Auditingdaten somit zunächst in ein passendes Format überführt werden; der tatsächliche Versand erfolgt dann über einen sicheren GSI-Transportkanal mittels HTTPS.

Die zentralen Sicherheitserfordernisse aus Abs. 4.2 sind somit ebenso erfüllt wie die Forderung nach Einhaltung relevanter Standards. Die Authentizität und Integrität der Auditing-Daten ist durch die verschlüsselte Übertragung sichergestellt, die von einem Dritten nicht ohne Kenntnis der privaten Schlüssel des Empfängers abgehört oder verändert werden kann. Durch die Festlegung auf einen in Globus-Infrastrukturen weit verbreiteten Standard für den strukturierten Versand und Empfang von Nachrichten wird auch der Standardkonformität der Lösung Rechnung getragen.

4.4.4 Der Auditing-Webservice

Um die von Grid-Ressourcen generierten Datensätze zu empfangen und weiterzuverarbeiten, war ein dedizierter Dienst zu entwerfen. Dieser mußte den oben formulierten Anforderungen genügen. Darüber hinaus waren jedoch noch weitere Bedingungen zu erfüllen, um eine sinnvolle Umsetzung zu erlangen. Seine Hauptaufgabe ist zum Einen die sichere Entgegennahme und Verifizierung von Auditing-Datensätzen, zum Anderen auch deren Aggregation und persistente Speicherung in einem Datenbank-Subsystem.

In Globus Toolkit 4 wurde erstmals der „Globus Container“ eingeführt (siehe Abs. 2.2.2), der als generische Serverumgebung für Grid-Dienste zur Verfügung steht. Es liegt nahe, sich für eine Auditing-Infrastruktur dieser generischen Serverlösung zu bedienen, um die Implementierung fehlerträchtiger und aufwendiger proprietärer

Server und Protokolle zu vermeiden. Der Rückgriff auf den Globus Container als Grundlage für den Auditing-Dienst genügt zudem den nichtfunktionalen Anforderungen der Standardkonformität und Interoperabilität, wie sie in Abs. 4.2 definiert wurden.

Durch die Verwendung des Globus Containers für den Auditing-Dienst werden auch die funktionalen Sicherheitsanforderungen erfüllt, denn der dem Globus Container zugrundeliegende Java WS Core (siehe Abs. 2.2.2) enthält eine GSI-Implementierung, auf die alle im Container gestarteten Dienste zurückgreifen können. Der sichere Empfang von Auditing-Datensätzen ist somit gewährleistet.

Da der Globus Container das WSRF-Framework implementiert, ist auch ein passender Kommunikationsstandard für den Datenaustausch mit den Ressourcen, die Auditing-Daten senden, gefunden. Mit für die Grid-Programmierung spezalisierten Entwicklungsumgebungen (siehe Abs. [FSF06]) sind die für das Auditing notwendigen Funktionen, insbesondere die Persistenzfunktionen, schnell zu implementieren.

Der Auditing-Dienst wurde als generischer Dienst entworfen, der im Rahmen eines übergreifenden Frameworks eingesetzt werden kann, um nicht nur Datensätze bezüglich Proxy-Credentials, sondern in Zukunft auch solche Daten entgegenzunehmen, die sich mit anderen zu auditierenden Subjekten befassen. Beim Proxy-Auditing handelt es sich um einen möglichen Anwendungsfall. w

Sicherer Empfang von Auditing-Datensätzen

Der Globus Container stellt über seine GSI-Implementation die Möglichkeit zur Verfügung, Nachrichten per HTTP über SSL (HTTPS) entgegenzunehmen. Er bedient sich dabei aller aus anderen GSI-Implementationen bekannten Mechanismen, also insbesondere der zertifikatsbasierten gegenseitigen Authentifizierung, verschlüsselter Kommunikation und der Möglichkeit einer Autorisierung über grid-mapfile (siehe Abs. 3.1.2). Mittels der GSI-Unterstützung des Globus Container kann jeder laufende Dienst über eine eigene Dienst-URL von außen angesprochen werden. Somit ist der authentische, vertrauliche Empfang von Auditing-Daten durch die Nutzung des Globus Containers sichergestellt.

Persistenz

Einmal empfangene Auditingdaten müssen dauerhaft gespeichert werden, um eine Analyse und Missbrauchserkennung a posteriori zu ermöglichen. Ein einzelnes Da-

tum ist für sich genommen nicht aussagekräftig genug, um bereits einen Missbrauchsverdacht zu begründen. Erst die Aggregation mit zuvor gespeicherten Informationen kann sinnschöpfend wirken. Somit war für den Auditing-Dienst ein Datenbank-Subsystem vorzusehen, das Auditingdaten speichert, aber auch eine strukturierte Abfrageschnittstelle zum schnellen Auffinden zuvor gespeicherter Daten zur Verfügung stellt.

Es lag nahe, hier auf eine relationale Datenbank (RDBMS) zurückzugreifen, die mittels SQL (Structured Query Language) abgefragt werden kann. Ein solches Datenbanksystem ist gut für die Speicherung der strukturierten, gleichförmigen Datensätze geeignet, die im Auditing erhoben werden. Desweiteren unterstützen praktisch alle relevanten Programmiersprachen RDBMS wie MySQL oder PostgreSQL, so dass eine Implementierung deutlich erleichtert wird. In Hinblick auf eine Visualisierung der Auditingdaten über ein webbasiertes Informationsportal (siehe Abs. 4.4.5) wird dieser Vorteil deutlich, da die derzeit für Webanwendungen üblicherweise verwendeten Sprachen wie etwa PHP, Perl oder auch Java über ausgereifte und leicht umsetzbare Anbindungen an relationale Datenbanksysteme verfügen.

Die Persistenzebene im Auditing-Dienst muss zweierlei Funktionen zur Verfügung stellen: Zunächst müssen neue Datensätze mit einer Eingabefunktion ans Datenbanksystem übergeben und dort gespeichert werden; desweiteren ist eine Abfragemöglichkeit für Auditing-Datensätze anhand definierter Kriterien wie Zertifikats-DN, Seriennummer oder zeitlicher Abfolge notwendig. Die Änderung einmal gespeicherter Auditing-Daten ist hingegen nicht möglich. Im Gegenteil: Auditing-Datensätze nachträglich zu ändern, würde das Auditingergebnis verfälschen.

Im Rahmen dieses umfassenden Auditing-Frameworks ist die Aufteilung der Daten in Tabellen hilfreich, da so die Auditing-Daten verschiedener Typen (Proxy-Credentials, Daten, Jobs etc.) voneinander separiert werden und in Tabellen verschiedener Struktur aufbewahrt werden können. So ist der Auditing-Dienst auch für zukünftige Anwendungen einsetzbar, die andere Datenprimitiva voraussetzen als das hier exemplarisch angeführte Proxy-Auditing.

Positionierung

In einer Grid-Infrastruktur steht der Auditing-Dienst nicht für sich alleine, sondern muss sich in eine umfangreiche Umgebung einfügen, in der bereits viele weitere Dienste und Server, teilweise zentralisiert und teilweise dezentral aufgestellt sind.

Um eine hohe Akzeptanz bei Nutzern und Administratoren zu erreichen, aber auch möglichst wenige Schwierigkeiten bei der Kommunikation mit Grid-Komponenten zu induzieren, muss die Positionierung des Auditing-Dienstes in der (teilweise virtuellen) Netzstruktur des Grid betrachtet werden.

Eine Grid-Infrastruktur wie das EGI kann in verschiedene administrativ-technische Zonen unterteilt werden, in denen ein Auditing-Dienst jeweils aufgestellt werden könnte (siehe Abb. 4.3).

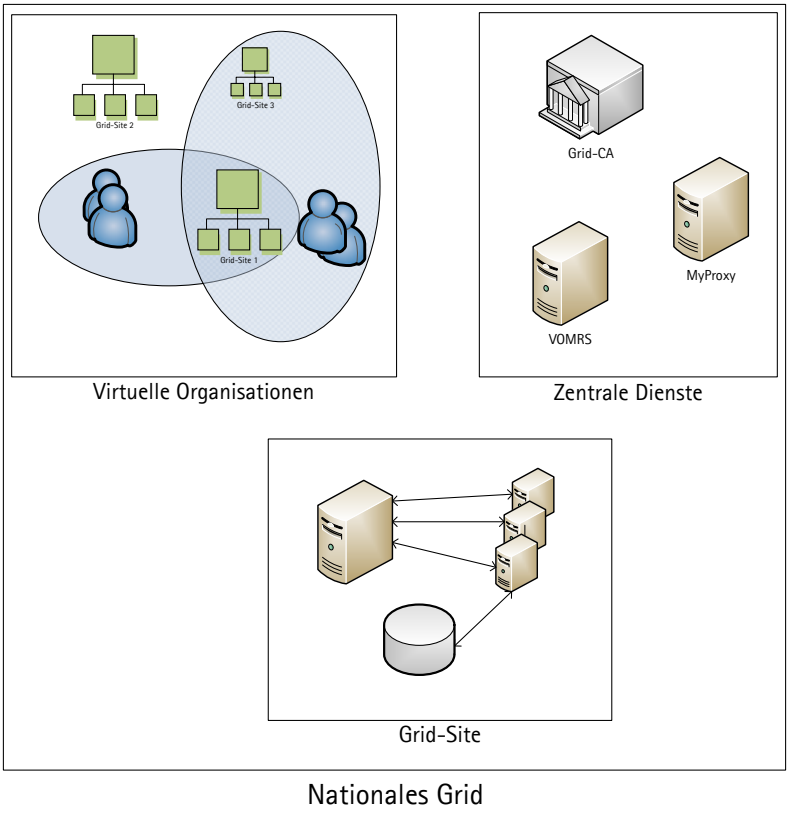


Abbildung 4.3: Administrative Zonen in einer nationalen Grid-Infrastruktur

Die unterste dieser Zonen ist die *„Grid-Site“*, also ein Cluster mit Daten- und Administrationsressourcen, der eine technische Einheit bildet und von einer einzelnen Institution administriert wird. Im D-Grid gibt es in wechselnder Zusammensetzung etwa 15 dieser Sites, die von verschiedenen Universitäten und Forschungseinrichtungen betrieben werden. Eine dieser Sites befindet sich etwa am Regionalen Rechenzentrum für Niedersachsen. Jede Grid-Site ist in der Regel in einem eigenen Netzwerkabschnitt zusammengefasst, der auf dem Zugangsnetz der Heimatorganisation basiert und durch eine Firewall gegen unberechtigte Zugriffe von Außen geschützt

ist, während die reibungslose Kommunikation mit anderen Grid-Sites oder -Nutzern sichergestellt ist. Am Übergang zwischen Grid-Site und Außenwelt ist neben der durch die Firewall gebildeten technischen Abgrenzung auch eine organisatorische Abgrenzung anzunehmen.

Die nächsthöhere Ebene ist die künstliche Ebene der *virtuellen Organisation*. Diese Ebene orientiert sich nicht an realen administrativen oder technischen Grenzen und wird ad hoc nach Bedarf zusammengestellt. So können Teile einer Grid-Site oder des Personals einer Grid-Site, aber auch ganze Grid-Infrastrukturen Teil einer virtuellen Organisation sein. In Abs. 2.2.1 werden virtuelle Organisationen näher beleuchtet.

Die Aggregation aller Grid-Sites eines Landes bildet die *nationale Grid-Infrastruktur*. Diese Infrastruktur (in Deutschland das D-Grid bzw. NGI-DE) wird oft zentral vom jeweiligen Forschungsministerium bezuschusst und bietet neben der Summe aller Teile (nämlich der Grid-Cluster und Datenressourcen) zentrale Dienste wie das VO-Management und die Certificate Authority.

In internationalen Grid-Föderationen wie der European Grid Initiative oder dem Baltic Grid sammeln sich verschiedene nationale Grid-Infrastrukturen und ermöglichen ihren Nutzern eine interoperable föderationsweite Nutzung, was nicht unerhebliche rechtliche, regulatorische und technische Probleme mit sich bringt.

In der Betrachtung, welche dieser Hierarchieebenen für die Positionierung des Auditing-Dienstes geeignet ist, spielen einige Faktoren eine Rolle. So ist zunächst der Datenschutz ein wichtiger Faktor. Da es sich bei Auditingdaten wie in Abs. 4.6 gezeigt um personenbezogene Daten handelt, sollten diese nicht ohne explizite Einwilligung des Inhabers an Stellen außerhalb des Erhebungslandes übertragen werden. Somit scheidet eine Platzierung des Auditing-Dienstes auf internationaler Ebene prinzipiell aus.

Es gilt in der D-Grid-Infrastruktur als gute Praxis, nur lange erprobte und von allen Nutzern benötigte Dienste zentral bereitzustellen (wie etwa den VOMRS, die CA, sowie das zentrale Monitoring), ein Auditingdienst jedoch wird in der Erprobungsphase nur von einem relativ kleinen Kreis genutzt. Daher wird eine Platzierung auf nationaler Ebene für diese erste Phase nicht avisiert.

Bei einem weiteren Grid-Dienst, der Proxy-Credentials verarbeitet, wie MyProxy (siehe Abs. 1.5.1), hat sich als übliche Praxis die Platzierung eines MyProxy-Servers in jeder Grid-Site herauskristallisiert. Dieses Vorgehen soll zunächst für

den Auditing-Dienst beibehalten werden, so dass jede Grid-Site für ihre Nutzer einen eigenen Dienst anbieten kann, wenn diese es wünschen. Da die Kommunikation über standardisierte Schnittstellen geschieht, die auch für die restliche Grid-Kommunikation verwendet werden (und somit in den Firewalls an Site-Grenzen freigeschaltet sind), ergeben sich aus dieser Platzierung keine technischen Schwierigkeiten. Der ordnungsgemäße Empfang von Auditingdaten ist auch dann gewährleistet, wenn der betreffende Nutzer auf einem Grid-Cluster außerhalb seiner Heimat-Site arbeitet.

Somit ergibt sich zunächst der Ansatz, den Auditing-Dienst auf Site-Ebene anzusiedeln. Eine spätere Verlagerung auf die nationale Ebene, um Auditing als gridweiten Dienst für alle angeschlossenen Sites anbieten zu können, ist ohne Weiteres möglich.

Redundanzen

Um zu vermeiden, dass Angreifer das Proxy-Auditing durch gezielte Angriffe auf den Auditing-Dienst lahmlegen, wird bereits in der Entwurfsphase eine grundlegende Möglichkeit zur Redundanz geschaffen. Nutzer, die von dieser Redundanz Gebrauch machen möchten, benötigen dazu eine Anmeldung auf mehreren Auditing-Diensten (etwa denen ihrer Heimatorganisation und einer nahegelegenen Grid-Site) und geben bei Erstellung ihres Proxy-Credentials mehrere URLs an. Die Auditing-Mechanismen in der GSI können dann nacheinander einen Versand der Auditing-Daten an all diese URLs versuchen und den ersten funktionierenden Dienst auswählen. So ist selbst bei einer erfolgreichen Attacke gegen den bevorzugten Auditing-Dienst noch ein Fallback möglich.

Kommunikation der Auditing-Dienste untereinander

Für die Synchronisation von Auditingdaten zwischen den Auditing-Diensten einer Grid-Infrastruktur können dieselben Mechanismen und Protokolle wie beim regulären Versand von Auditingdaten genutzt werden. Über eine WSRF-gestützte sichere Kommunikationsschnittstelle können die Dienste untereinander Daten austauschen. Dieser Datenaustausch wird insbesondere dann notwendig, wenn ein Auditingdienst aufgrund eines Angriffs oder eines technischen Problems zeitweise nicht verfügbar war und die Auditingeinträge an einen alternativen Dienst gesendet wurden. Dieser

sollte dann baldmöglichst eine Synchronisation mit dem ursprünglich angefragten Auditingdienst durchführen.

4.4.5 Nutzer-Interaktion mit dem Auditingdienst

Weite Teile der Infrastruktur zum Proxy-Auditing kommen ohne direkte Interaktion mit dem Endnutzer aus. Der gesamte Vorgang des Auditings, also die Erhebung, der Versand und die Auswertung von Auditing-Einträgen, findet ohne Zutun des Benutzers statt. Dieser muss lediglich bei der Erstellung eines Proxy-Zertifikats die notwendigen Vorkehrungen für das Auditing treffen und kann – sofern gewünscht – die gesammelten Daten später einsehen.

Im Rahmen eines Frameworks zum Auditing verschiedener Aspekte der Grid-Nutzung können diese Interaktionsmöglichkeiten noch um Auswahlmöglichkeiten zu Auditing-Subjekten (also Datenbeständen, bestimmten Grid-Jobs etc.) erweitert werden; diese Dissertation betrachtet jedoch ausschließlich das Proxy-Auditing.

Interaktive Zertifikatserzeugung

Der Nutzer erstellt ein Proxy-Zertifikat zur Nutzung im Grid in der Regel mittels eines Webportals oder des Kommandozeilenprogramms `grid-proxy-init`, das im Lieferumfang des Globus Toolkit enthalten ist. Als Teil der Zertifikatserstellung kann der Nutzer eine Option an das Kommandozeilenwerkzeug übergeben, um eine Liste mit URLs zu Auditing-Diensten im Zertifikat zu integrieren. Der Signaturvorgang des Proxy-Zertifikats umfasst dann auch die in Abs. 4.4.2 beschriebene Zertifikatserweiterung. Ist diese erfolgreich in das signierte Proxyzertifikat integriert, ist für den Nutzer keine weitere Interaktion mit dem Auditing-System mehr notwendig, bis er die gesammelten Daten einsehen möchte oder über das integrierte Warnsystem vor möglichem Missbrauch gewarnt wird.

Datenansicht über Webportal

Bei einem nutzerzentrischen System wie der hier vorgestellten Auditing-Infrastruktur spielt die Transparenz gegenüber dem Benutzer selbstverständlich eine große Rolle. Daher wird als weitere Interaktionsmöglichkeit des Nutzers mit dem Auditing-System ein Webportal entworfen, über jenes Informationen zur Nutzung von Proxy-Credentials und mögliche Missbrauchsfälle eingesehen werden können. Durch die

Entscheidung, alle anfallenden Datensätze in einem relationalen Datenbanksystem vorzuhalten, wird die Umsetzung eines derartigen Portals deutlich erleichtert.

Die Grundfunktion des Webportals ist die Ansicht aggregierter Auditingdaten, die auf einem vom Nutzer mit entsprechend Abs. 4.4.5 erstellten Proxy-Credential basieren. Diese können dann anhand weiterer Kriterien wie etwa der zeitlichen Abfolge, mithilfe topologischer Informationen jedoch auch in Form einer „Grid-Landkarte“ dargestellt werden. Eine Modifikation der vorhandenen Daten darf jedoch über ein Webportal (wie in Abs. 4.4.4 erwähnt) nicht erfolgen, um nachträgliche Manipulationen und die damit einhergehenden Inkonsistenzen zu vermeiden.

Da das Webportal persönliche Daten der Nutzer zur Ansicht bereitstellt, ist es gegen unbefugten Zugriff abzusichern. Der Zugriff auf das Portal sollte somit ausschließlich über eine authentische, vertrauliche Verbindung per SSL/TLS und nach erfolgter Authentifizierung ermöglicht werden. Hier bietet sich eine Authentifizierung mittels desjenigen Zertifikats an, für dessen „Nachkommen“ Auditingdaten erhoben werden. Eine Authentifizierungsmöglichkeit über digitale Zertifikate ist in modernen Web-Applikationsservern, die SSL/TLS unterstützen, problemlos möglich.

Missbrauchswarnung

Um dem Nutzer die Möglichkeit zu geben, auf vermuteten Missbrauch seines Proxy-Credentials adäquat zu reagieren, ist eine bloße Anzeige mittels eines Webportals unter Umständen nicht ausreichend. Da mittels der in Kapitel 6 vorgestellten Komponente zur Missbrauchserkennung eine zuverlässige Einordnung der Credentialnutzung möglich ist, kann eine zusätzliche Dienstleistung des Auditingsystems darin bestehen, den Benutzer automatisch zu warnen, sobald es einen potentiellen Missbrauch ausgemacht hat. Diese Warnung kann zweckmäßigerweise per E-Mail an die im X.509-Zertifikat des Nutzers hinterlegte Adresse erfolgen. Somit ist sichergestellt, dass Missbrauchswarnungen den korrekten Adressaten erreichen.

4.5 Angriffe auf das Auditing-System

Erfahrungen aus dem Security Engineering zeigen, dass Systeme zum Tracking und Logging von Angreifern als sekundäre Ziele gewählt werden, um nach erfolgreichem Angriff des Primärziels Spuren zu verwischen. Ist das Logging und Tracking eines

IT-Systems außer Gefecht gesetzt, wird die (Straf-)Verfolgung vergangener und zukünftiger Attacken entscheidend erschwert.

Angriffe gegen das Auditing-System sollten also in der Evaluation berücksichtigt werden; wenn ein System zur Missbrauchserkennung allzuleicht ausgetrickst werden kann, ist ein echter Sicherheitsgewinn nicht mehr gewährleistet. Im Folgenden werden einige denkbare Angriffsszenarien evaluiert, die sich gegen die Auditing-Infrastruktur richten und deren Deaktivierung zum Ziel haben.

In Tabelle 4.1 findet sich eine Übersicht möglicher Angriffe gegen das Auditing und ihrer Abwehrbarkeit durch das System.

Angriff	Angriff abwehrbar
Entfernen der Zertifikatserweiterung	✓
Störung des Auditing-Dienstes	✓
Unterbinden der Kommunikation zwischen Host und Auditing-Dienst[1]	
Manipulation der GSI-Bibliotheken[1]	

Tabelle 4.1: Mögliche Angriffe gegen die Auditing-Infrastruktur

Cracking des Auditing-Dienstes

Angriffe, die darauf abzielen, den Auditing-Dienst selber zu übernehmen (Cracking), werden im Folgenden ausgeklammert. Diese Annahme fußt auf zwei Gründen. Zum Einen wird im vorgestellten Konzept der Auditing-Dienst stets auf einem eigenen, abgetrennten Hostsystem betrieben und gegen Angriffe mit Mitteln des Betriebssystems besonders abgesichert (gehärtet). Diese Härtung ist für einzelne und relativ alleine stehende Dienste auch in einer Grid-Umgebung machbar, für das Gros der Komponenten (insbesondere derer, die direkt von Nutzern bedient werden) jedoch meist zu aufwendig. Da der Auditing-Dienst zum Anderen lediglich eine einzige

[1] Administrative Privilegien auf dem angegriffenen Server notwendig

Schnittstelle nach außen anbieten muss – den Auditing-Webservice in einem Globus-Container –, ist das Potenzial für einen erfolgreichen Angriff sehr gering. Für den Globus Container existieren keine bekannten Lücken, die eine Übernahme des Dienstes durch Angreifer erlauben[2]. Ein erfolgreicher Angriff gegen den Auditing-Dienst ist somit sehr unwahrscheinlich.

Entfernen der Zertifikatserweiterung

Das Auditing von Proxy-Credentials setzt das Vorhandensein einer Erweiterung im vom Nutzer erzeugten ersten Proxy-Zertifikat voraus. Gelänge es einem Angreifer, diese zu entfernen ohne das Zertifikat ungültig zu machen, so wäre damit das Auditing außer Kraft gesetzt.

In der Praxis gibt es jedoch keine Möglichkeit, die Zertifikatserweiterung zu entfernen. Würde ein Angreifer die Erweiterung aus den Metadaten löschen, so würde zwangsläufig die digitale Signatur und somit das gesamte Zertifikat ungültig. Um dessen Signatur zu erneuern, ist jedoch der private Schlüssel des Nutzerzertifikats (EEC) notwendig – und dieser ist für Angreifer nicht zugänglich.

Würde ein Angreifer hingegen versuchen, das erste abgeleitete Proxy-Zertifikat aus der an Grid-Ressourcen übermittelten Kette zu entfernen und eine andere Ableitung für seine Zwecke zu missbrauchen, so wäre die „Chain of Trust“ ungültig und eine erfolgreiche Kommunikation mit GSI-gestützten Ressourcen ebenfalls unmöglich.

Eine Entfernung oder Veränderung der X.509-Extension ist somit kein probates Mittel, um das Auditing zu verhindern.

Störung des Auditing-Dienstes – Denial of Service

Die Ausschaltung eines Dienstes durch absichtliche Überlastung (Denial of Service, kurz DoS) ist eine häufig gewählte Angriffsart, die – sofern dem Angreifer ausreichende Ressourcen, etwa in Form eines Botnets, zur Verfügung stehen – auf nicht redundante Systeme eine verheerende Wirkung haben kann.

Ein Angreifer, der die Auditing-Infrastruktur überlasten will, könnte dies im einfachsten Fall mit massenhaften sinnlosen Anfragen an den Auditing-Dienst bewerkstelligen (etwa einer „Syn Flood“). Gegen derlei Angriffe bietet der Netzwerkstack

[2]Eine Anfrage bei den Entwicklern des Globus Toolkit im Februar 2011 wurde durch Ian Foster beantwortet. Ihm selber seien keine Exploits bekannt.

vieler Betriebssysteme einen rudimentären Schutz; eine Überlastung der Netzwerkanbindung durch einen verteilten Angriff (distributed Denial of Service) lässt sich jedoch hierdurch nicht verhindern.

In der Konzeption der Auditing-Infrastruktur wurden Denial-of-Service-Angriffe jedoch berücksichtigt und eine einfache, jedoch effektive Gegenmaßnahme eingeführt. Die in Abschnitt 4.4.2 detailliert erläuterte X.509-Zertifikatserweiterung kann mehrere verschiedene URLs enthalten, die jeweils einen Auditing-Dienst adressieren. Ist der erste derart adressierte Dienst (etwa wegen einer DoS-Attacke) nicht erreichbar, wird nach einer konfigurierbaren Zeitspanne (Timeout) die nächste URL verwendet – so lange, bis ein Auditing-Eintrag erfolgreich versendet werden konnte.

Eine weitere Möglichkeit für Angreifer, einen „Denial of Service“ zu erzeugen, besteht darin, den Auditing-Dienst mithilfe eines bereits abgefangenen Proxy-Credentials mit syntaktisch korrekten, jedoch unsinnigen Auditing-Records zu überfluten, um so die tatsächlichen Angriffe zu „maskieren“. Diese Methodik würde jedoch aufgrund der für legitime Nutzung untypischen Muster unweigerlich als Angriff erkannt werden und somit ihren Zweck – nämlich die Behinderung der Angriffserkennung – verfehlen.

Angriffe mit „Denial of Service“-Charakteristiken können somit wirksam abgewehrt werden und stellen keine Gefahr für den Auditing-Dienst dar.

Unterbrechung der Kommunikation

Hat ein Angreifer eine Grid-Ressource unter seine Kontrolle gebracht, kann er seine erhöhten Privilegien dazu nutzen, die Kommunikation zwischen dieser Ressource und dem Auditing-Dienst zu verhindern. Das ist für Administratoren eines Systems trivial einfach möglich, indem lokal vorhandene Paketfilter (wie etwa netfilter/iptables[3] oder ipfw[4]) genutzt werden, um TCP-Verbindungen zu den Auditing-Diensten zu stören. Ein ähnliches Vorgehen legt Malware auf Client-Systemen oft an den Tag, um die Kommunikation zwischen dem infizierten System und den Signaturservern von Virenscannern zu unterbinden. Während der Auditing-Dienst nicht in der Lage ist, eine solche Störung zu bemerken, da ja keine Kommunikation bei ihm ankommt, würden Versuche, die Zertifikatsnutzung auf dem betroffenen System zu auditieren, mit Zeitüberschreitungen wegen Nichterreichbarkeit quittiert werden.

[3]Netfilter-Projekthomepage: `http://www.netfilter.org/`

[4]Dokumentation zu ipfw: `http://www.freebsd.org/doc/handbook/firewalls-ipfw.html`

Attacken, die die Kommunikation eines Grid-Knotens mit dem Auditing-Dienst unterbinden, lassen sich einerseits oft durch die stark erhöhten Abarbeitungszeiten (wegen der auftretenden Timeouts) recht leicht erkennen, andererseits durch eine Erweiterung des Auditing-Dienstes abwehren. Wenn der erfolgreiche Versand eines Auditing-Datensatzes (inklusive Quittierung durch den Auditing-Dienst) zur Voraussetzung jeder GSI-gestützten Kommunikation gemacht würde, könnte ein Angreifer selbst dann keinen Nutzen mehr aus unrechtmäßig erworbenen Grid-Proxy-Credentials ziehen, wenn er die Kommunikation der Grid-Ressource kontrolliert, auf der sie abgelegt sind. Es ergäben sich jedoch Implikationen für die Kompatibilität und Verträglichkeit des Auditing-Systems mit dem Grid, die zu berücksichtigen sind.

Manipulation der GSI-Bibliotheken

Die wirksamste Möglichkeit, das Auditing auf Grid-Ressourcen zu unterbinden, ist der Austausch der modifizierten, auditing-fähigen GSI-Bibliotheken durch solche, die nicht auditingfähig sind (also etwa jene Bibliotheken, die im Auslieferungszustand des Globus Toolkit enthalten sind). Ein Angreifer, der eine Grid-Ressource unter seine Kontrolle gebracht hat, könnte diesen Austausch mithilfe unrechtmäßig erworbener Administrator-Privilegien vornehmen, würde sich dabei aber einem hohen Risiko der Entdeckung aussetzen. Schließlich müssten für einen wirksamen Austausch Serverkomponenten neugestartet und ggf. signaturbasierte Mechanismen zur Erkennung veränderter Dateien (wie etwa tripwire[5]) ausgeschaltet werden. Hat er diese Hürden überwunden, steht ihm jedoch eine Grid-Ressource zur Verfügung, deren Kommunikation nicht auditiert wird und die somit als Sprungbrett für weitere Angriffe oder als unauffällige Senke für gestohlene Proxy-Credentials dienen kann.

Da die oben für den Auditing-Dienst angenommene Sicherheit gegen Angriffe von außen für andere Grid-Ressourcen nicht ohne Weiteres angenommen werden kann, ist die beschriebene Attacke denk- und machbar und sollte in der weiteren Konzeption berücksichtigt werden. Die oben genannten Gegenmaßnahmen finden hier keine Anwendung, da nicht der Auditing-Dienst angegriffen wird, sondern die Client-Komponenten.

Unter Vernachlässigung der Rückwärtskompatibilität kann diese Gefahr gebannt werden, indem die Auditing-Erweiterung im ersten Proxyzertifikat als „kritisch“ markiert wird und somit jede Komponente, die das Zertifikat verarbeitet, diese kri-

[5]Tripwire-Projekthomepage: `http://sourceforge.net/projects/tripwire/`

tische Erweiterung beachten muss. Kommunikationspartner, die nicht (oder nicht mehr) auditingfähig sind, können somit das Zertifikat nicht erfolgreich verarbeiten und müssen die Kommunikation abbrechen.

Da nicht rückwärtskompatible Änderungen an der Grid-Infrastruktur im vorliegenden Konzept jedoch vermieden werden sollen, musste eine andere Lösungsmöglichkeit gefunden werden. Diese existiert in Form eines zusätzlichen Moduls für das Auditing-System (siehe Abs. 5.7).

4.6 Datenschutz und Gesetzeslage

Durch den in den vergangenen Abschnitten beschriebenen Auditing-Dienst werden personenbezogene Daten erhoben, gespeichert und verarbeitet. Er fällt damit, zumindest in Deutschland, unter den Bereich der Datenschutz-Gesetzgebung. Das Bundesdatenschutzgesetz [Ver90] definiert personenbezogene Daten als *Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener)* (§3 Abs. 1). Ihre Verarbeitung, definiert als *die Erhebung, Verarbeitung oder Nutzung personenbezogener Daten unter Einsatz von Datenverarbeitungsanlagen* (§3 Abs. 2) unterliegt besonderen Regulierungen und ist nur unter bestimmten Bedingungen statthaft. Der §4 des BDSG legt diese Bedingungen fest: So dürfen personenbezogene Daten verarbeitet werden, wenn eine Rechtsvorschrift dies voraussetzt oder der Betroffene eingewilligt hat.

Da, wie in Abs. 4.3 erläutert, der Nutzer stets durch die Auswahl der entsprechenden Option bei der Erstellung eines Proxy-Credentials sein Einverständnis mit dem Auditing erklären muss, kann von der Einwilligung des Betroffenen regelmäßig ausgegangen werden. Aus datenschutzrechtlicher Sicht ergeben sich somit keine Probleme für das Auditing-System.

4.7 Zusammenfassung

Im vorliegenden Kapitel wurde ein umfassendes Framework für das Auditing in globus-basierten Grid-Infrastrukturen vorgestellt und seine grundlegenden Elemente erläutert. Zunächst wurden die nichtfunktionalen Anforderungen an dieses Framework erhoben. Dabei standen Kriterien wie die Interoperabilität mit anderen Grid-Komponenten und natürlich die Sicherheit des Gesamtsystems im Vordergrund.

Der Entwurf des Auditing-Systems sieht folgende Teilkomponenten vor:

1. Modifikationen der Globus-Komponenten, um Auditing-Informationen zu versenden
2. ein auf Webservice-Technologien aufgesetzter Dienst zum Empfang der Auditing-Informationen
3. eine relationale Datenbank zur Speicherung von Auditing-Daten
4. eine Komponente zur automatischen Missbrauchserkennung anhand gespeicherter Auditingdaten
5. ein Nutzer-Interface zur Einsicht gespeicherter Daten und zur Erzeugung auditing-kompatibler Proxyzertifikate

4.7.1 Fazit

Das vorgestellte Auditing-Framework unterstützt die Erstellung und den Versand von Auditing-Einträgen mittels einer modifizierten Version der GSI-Bibliotheken. Diese Modifikation wurde gewählt, um eine möglichst breite Unterstützung verschiedener Globus-Teilkomponenten zu erreichen, ohne das Proxy-Auditing jeweils auf Applikationsebene entwerfen, implementieren und testen zu müssen. Mit der Modifikation auf GSI-Ebene wird zudem sichergestellt, dass die notwendigen Informationen über Zertifikatsinhaber, -erweiterungen und weitere Metadaten programmatisch ausgelesen werden können.

Um Auditingdaten auf eine sichere und standardkonforme Art und Weise entgegenzunehmen, wurde ein auf dem durch Globus unterstützten WSRF-Standard basierender Webservice entworfen. Dieser stellt Authentizität und Vertraulichkeit der entgegengenommenen Einträge durch die Verwendung der GSI sicher und speichert sie in einer Persistenzschicht. Diese Persistenz wird durch eine relationale Datenbank sichergestellt, die durch ihr Tabellenkonzept verschiedenartige Daten ohne große Modifikation in der Programmlogik entgegennehmen kann.

Damit Nutzer mit dem Auditing-System adäquat interagieren können, wird zunächst eine Schnittstelle zur Erstellung spezieller Proxy-Zertifikate geschaffen. Diese Zertifikate enthalten eine X.509-Erweiterung, die von auditingfähigen Globus-Komponenten ausgewertet wird. In dieser Erweiterung befindet sich die URL des-

jenigen Auditing-Webservice, der die zum Zertifikat gehörenden Auditing-Einträge entgegennimmt.

Da diese Zertifikatserweiterung vom Inhaber digital signiert wird, ist seine Anweisung zum Auditing durch Dritte nicht manipulierbar. Ein Webportal zur Information über gesammelte Auditingdaten sowie ein Warnsystem geben dem Nutzer zudem die Möglichkeit, sich zeitnah über möglichen Missbrauch seiner Proxy-Credentials zu informieren.

Angriffe gegen den Auditing-Dienst und gegen die Auditing-Infrastruktur wurden im Entwurf berücksichtigt und können unter Wahrung der vollen Kompatibilität abgewehrt werden. Eine „Watchdog“-Komponente hilft bei der Abwehr komplexerer Angriffe gegen das Auditing.

Die automatisierte Erkennung missbräuchlicher Nutzung von Proxy-Credentials macht eine zusätzliche Teilkomponente notwendig, die anhand eines zu definierenden Regelwerks oder anderer Methoden des maschinellen Lernens ohne Zutun des Nutzers eine Einordnung der Audit Records vornimmt. Diese Teilkomponente ist Gegenstand des Kapitels 6.

Kapitel 5

Implementierung

Um die Umsetzbarkeit des im vorigen Kapitel vorgestellten Systems unter Beweis zu stellen, wurde eine Referenzimplementation der Auditing-Infrastruktur für die webservice-basierten Komponenten des Globus Toolkit durchgeführt. Diese Implementation setzt Entwurfsentscheidungen und Anforderungskatalog zu einem prototypischen Auditing-System für das Globus Toolkit 4 um.

Die in diesem Kapitel beschriebene beispielhafte Implementation wurde durch eine am Institut für Rechnernetze und verteilte Systeme angefertigte und vom Autor dieser Dissertation betreute Masterarbeit [Szo09] unterstützt.

5.1 Einleitung

Ein Auditing-Framework für das Grid, wie es im vorigen Kapitel vorgestellt wurde, löst ein grundlegendes paradigmenbedingtes Problem in Grid-Infrastrukturen: Es schafft auf Wunsch des Nutzers Transparenz, wo zuvor das Grid als „Black Box“ verwendet wurde. Das grundlegende Konzept und der Entwurf einer umfassenden Auditing-Lösung für das Grid wurden im vorigen Kapitel 4 ausführlich erläutert. Die praktische Umsetzbarkeit läßt sich anhand dieser theoretischen Überlegungen jedoch nicht adäquat feststellen. Daher wurde eine Referenzimplementation für die Java-Komponenten des Globus Toolkit 4 konzipiert und erstellt.

Die Beispielimplementation wurde in Java und für die Java-Komponenten des GT4 entwickelt, da diese die Mehrheit der in einer globus-basierten Infrastruktur verwendeten Dienste stellen. So wird der WebService-GRAM (kurz WS-GRAM), aber auch ein Dienst zum Datenmanagement (OGSA-DAI) sowie verschiedene Community-Dienste mithilfe des im Webservice-Framework des Globus Toolkit enthaltenen Globus Containers (siehe Abs. 2.2.2) betrieben.

In der Implementation wurden verschiedene Teile des Globus Toolkit modifiziert, um die für das Auditing notwendigen Voraussetzungen zu schaffen:

1. Die Werkzeuge zum Erstellen neuer Proxy-Credentials (`grid-proxy-init`) wurden angepasst, um die X.509-Zertifikatserweiterung zum Auditing in ein Proxy-Zertifikat einfügen zu können.

2. Die GSI-Funktionen des Java WS-Core wurden modifiziert, um auditingfähig zu werden.

3. Ein WSRF-Webservice auf Basis des Globus Containers wurde erstellt, der Auditingdaten entgegennimmt und speichert.

Zur Veranschaulichung sei auf Abbildung 5.1 hingewiesen. Hier wird das Zusammenspiel und der Kommunikationsfluss der einzelnen Komponenten in Anlehnung an Abb. 4.1 nochmals illustriert.

5.2 Vorbereitung und Versand

Auditing-Informationen sollen auf verschiedenen Ressourcen, aber in einem einheitlichen Format und auf eine einheitliche Art und Weise gesammelt werden, um ei-

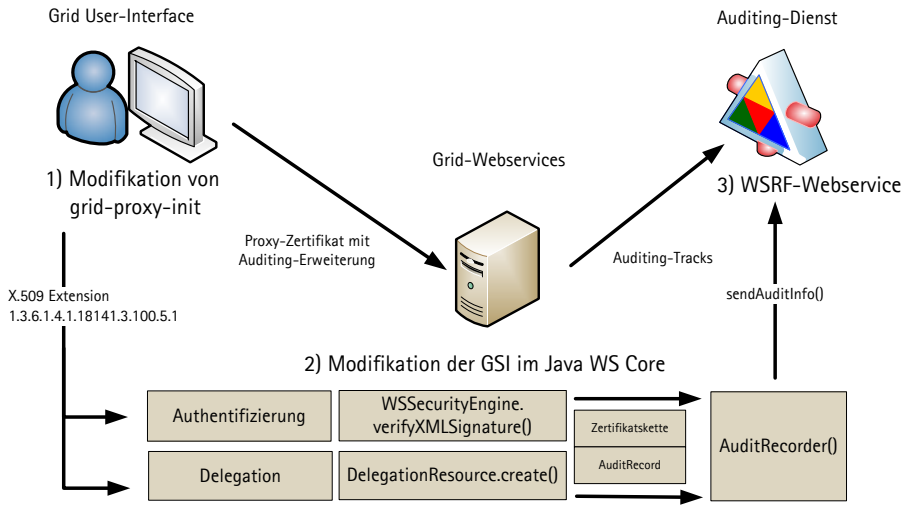


Abbildung 5.1: Zusammenspiel der modifizierten Komponenten in der Implementation

ne konsistente Verarbeitung durch den Auditing-Dienst zu gewährleisten. In der Referenzimplementation wurde zu diesem Zweck die Klasse `AuditRecorder` implementiert. Diese Klasse wird an den programmatisch geeigneten Stellen (siehe den folgenden Abschnitt 5.3) instanziiert und kapselt die Zusammenstellung und den Versand der Auditing-Records.

Dem Konstruktor der Klasse muss die Zertifikatskette des aktuell zu auditierenden Credentials, der aktuelle Nutzungsmodus (Authentifizierung oder Delegation) und sofern vorhanden, der Name des Dienstes, der das zu auditierende Credential verwendet, übergeben werden (siehe Listing 5.1). Die Zertifikatskette liegt als Array aus Zertifikatsobjekten vor, dessen oberstes Element das letzte abgeleitete Proxy-Zertifikat ist; das End-Entity-Zertifikat befindet sich folgerichtig am unteren Ende des Arrays. Da die Chain of Trust zum Zeitpunkt des Auditing bereits verifiziert ist, ist das CA-Zertifikat nicht Bestandteil der Zertifikatskette.

Wird nun die Methode `sendAuditInfo()` aufgefrufen, so wird das in Tabelle 5.1 mit `certChain[n-1]` bezeichnete erste Proxy-Zertifikat nach der Auditing-Erweiterung durchsucht. Wird diese Erweiterung nicht im ersten Proxy-Zertifikat gefunden, bricht der `AuditRecorder` die Bearbeitung ab, da nur das vom Nutzer direkt signierte Proxy-Zertifikat als gültige Willenserklärung für oder gegen das Auditing dient. Eventuell vorhandene Auditing-Erweiterungen in anderen Delegationen dürfen und werden nicht ausgewertet.

`certChain[n]`	EEC des Benutzers
`certChain[n-1]`	direkt vom EEC abgeleitetes Proxy-Zertifikat (beim Aufruf von `grid-proxy-init`)
...	
`certChain[1]`	weiteres abgeleitetes Zertifikat
`certChain[0]`	weiteres abgeleitetes Zertifikat

Tabelle 5.1: Aufbau der Zertifikatskette im certChain-Objekt (Quelle: [Szo09])

Sofern die Auditing-Erweiterung gefunden wurde, wird die enthaltene URL extrahiert und gespeichert. Sodann ermittelt der AuditRecorder den Distinguished Name des vom aufrufenden Dienst verwendeten Hostzertifikats; hierzu wird die Globus-SecurityManager-Klasse verwendet. Um die für einen Auditing-Record notwendigen Daten zu komplettieren, wird dann die Seriennummer, die Aussteller-Seriennummer und der Distinguished Name des aktuell zu auditierenden Zertifikats (das ist in aller Regel das oberste Element des `certChain`-Arrays) abgefragt und gespeichert. Die aktuelle Systemzeit wird mit einem Aufruf von `System.currentTimeMillis()` ermittelt und die gesammelten Informationen mittels der Methode `sendToCLMS()` versandt. Der Versand erfolgt mittels GSI Transport (siehe 2.2.3).

```
AuditRecorder recorder = new AuditRecorder(certChain,
                                           AuditRecord.
                                              USAGE_MODE_DELEGATION,
                                           "GT4 Delegation Service");
recorder.sendAuditInfo();
```

Listing 5.1: Instanziierung und Auditing mit dem AuditRecorder

5.3 Modifikation des Java WS-Core

Eine grundlegende Aufgabe der Referenzimplementation ist es, diejenigen Stellen in den Webservice-Basisbibliotheken des Globus Toolkit zu identifizieren, an denen der in Abschnitt 5.2 eingeführte Auditing-Aufruf vorgenommen werden muss.

Für das Auditing von Authentifizierung und Delegation neuer Credentials müssen zudem unterschiedliche Komponenten modifiziert werden.

5.3.1 Authentifizierung

Die Authentifizierung von Nutzern gegenüber Komponenten kann, wie in Abs. 2.2.3 beschrieben, mittels *GSI Secure Message*, *GSI Secure Conversation* und *GSI Transport* erfolgen. Alle drei Authentifizierungsprotokolle wurden modifiziert, um das Auditing von Authentifizierungsvorgängen zu ermöglichen.

Das Kommunikationsschema *GSI Secure Message* beinhaltet eine Verschlüsselung der Nutzdaten in einer Nachricht (Message-Level Security) und basiert auf SOAP-Nachrichten, die gemäß dem WS-Security-Standard [NKMHB06] behandelt werden. Das Auditing greift hier an der Stelle ein, an der die relevanten Informationen aus einem SOAP-Header extrahiert und überprüft werden. Nachdem eine per GSI Secure Message gesicherte Nachricht beim Empfänger eingetroffen ist, wird diese an eine Instanz der `WSSecurityEngine`-Klasse weitergegeben. In dieser wird die Nachricht mittels der Methode `processSecurityHeaders` auf die Existenz einer digitalen Signatur und verschlüsselter Daten überprüft. Dazu wird die Methode `verifyXMLSignature` ausgeführt, die zunächst die Zertifikatskette aus dem SOAP-Header extrahiert und dann die Signatur der Nachricht prüft.

Nach der Signaturprüfung wird die Chain of Trust der Zertifikatskette untersucht - nachdem auch diese Prüfung erfolgreich abgeschlossen wurde, ist die Authentifizierung des Nutzers abgeschlossen. An dieser Stelle setzt das Auditing an und macht sich zunutze, dass die Zertifikatskette noch aus dem zuvor abgeschlossenen Vorgang zur Verfügung steht. Da programmatisch kein eindeutiger Service-Name ermittelt werden kann, wird im Auditing-Eintrag die aufgerufene URL des Nachrichtenkontextes verwendet (mittels des Methodenaufrufs

`msgCtx.getProperty("transport.url")`).

Findet die Kommunikation mittels GSI Secure Conversation statt, so ist der sog. `SecurityContextHandler` für sämtliche sicherheitsrelevanten Protokollfunktionen zuständig. Er prüft bei eingehenden Verbindungen, ob ein bereits bestehender Sicherheitskontext zur Verfügung steht; wenn nicht, wird ein neuer Kontext des Typs `GSSContext` instanziiert. Der Ansatzpunkt für das Auditing ergibt sich, analog zur Implementation im vorigen Fall, während der Prüfung der Credentials eines neuen Sicherheitskontexts. Nachdem der GSI-Handshake erfolgreich war und die Zertifikatskette verifiziert wurde, kann sie an den `AuditRecorder` übergeben und ein Auditing-Datensatz versandt werden. Da auf dieser tiefen Protokollebene keine

Servicenamen oder -URLs vorliegen, wird hier zunächst als Dienstbezeichnung ein Platzhalter eingesetzt, bevor der Auditing-Datensatz übermittelt wird.

Kommunikationssicherheit mittels *GSI Transport* basiert auf HTTPS und wurde erweitert, um die Zertifikatskette von Proxy-Zertifikaten zu verifizieren. Analog zum Ablauf bei GSI Secure Conversation wird hier zunächst ein Sicherheitskontext aufgebaut und dann die validierte Zertifikatskette dem `AuditRecorder` zur Verfügung gestellt. Auf eine Dienst-URL oder -Bezeichnung kann jedoch auch bei GSI Transport nicht zurückgegriffen werden, so dass lediglich die im Host- und Proxy-Zertifikat zur Verfügung stehenden Informationen in den Auditing-Record einfließen.

Mit je einer Auditing-Implementation für GSI Secure Message, Conversation und Transport werden somit alle in GT 4.0 verwendeten Authentifizierungsmechanismen vom Auditing abgedeckt.

5.3.2 Delegation

In den Webservice-basierten Komponenten des Globus Toolkit 4.0 können Delegationen entweder über einen WSRF-Webservice, den „Delegation Service“, oder als Teil einer „GSI Secure Conversation“ (beide werden in Abs. 2.2.3 beschrieben) durchgeführt werden.

Der *Delegation Service* speichert alle von ihm verwalteten delegierten Credentials in WSRF-Ressourcen. Das Auditing wird hier integriert, indem die Erzeugung neuer Ressourcen an den Auditing-Service gemeldet wird. Diese Erzeugung findet in der Methode `DelegationResource.create()` statt; eine Instanz des `AuditRecorder` wird demnach hier erzeugt.

Werden Delegationen über die „*GSI Secure Conversation*“ gebildet, ist der Delegation Service ausgeklammert; hier muss also eine zusätzliche Modifikation für das Auditing stattfinden. Sobald nach erfolgreichem Handshake zwischen Server und Client ein Security-Kontext gebildet wurde, generiert der Server ein Schlüsselpaar und einen CSR, die an den Client zum Signieren übertragen werden. Die so entstandene Delegation wird dann zurück an den Server übertragen, der sie verifiziert und als Globus-Credential abspeichert.

Nachdem dieser Delegationsprozess erfolgreich abgeschlossen wurde, wird in der modifizierten Implementation eine Instanz des `AuditRecorder` gebildet. Dieser analysiert die zuvor vom Server abgespeicherte Zertifikatskette inklusive des neuen, zu auditierenden Proxy-Credentials und versendet gegebenenfalls einen Auditing-

Datensatz. Zwei weitere mit der Delegation von Credentials befasste Methoden, `initDelegation` und `acceptDelegation`, wurden analog modifiziert.

Somit werden alle Delegationsprozesse, die im webservice-basierten Teil des Globus Toolkit erfolgen, durch das Auditing überwacht.

5.4 Modifikation der Grid-Proxy-Tools

Eine Anforderung im Entwurf der Auditing-Infrastruktur (Abs. 4.3 und 4.4.2) war die Möglichkeit für den Nutzer, das Auditing selbstständig zu (de-)aktivieren. Desweiteren benötigen die modifizierten GSI-Bibliotheken eine nicht veränderbare Ziel-URL, an die alle Auditing-Einträge gesendet werden. Beide Anforderungen werden in einer Erweiterung für Proxyzertifikate zusammengefasst, die im Detail in 4.4.2 beschrieben ist.

5.4.1 Zertifikatsextension

Ein Zertifikat muss verschiedene Pflichtangaben enthalten, die allesamt vom Kommandozeilenwerkzeug `grid-proxy-init` bereits in ein neues Proxycredential eingefügt werden. Der X.509-Standard [ITU05] definiert ein digitales Zertifikat wie folgt:

```
Certificate ::= SIGNED { SEQUENCE {
 version [0] Version DEFAULT v1,
 serialNumber CertificateSerialNumber,
 signature AlgorithmIdentifier,
 issuer Name,
 validity Validity,
 subject Name,
 subjectPublicKeyInfo SubjectPublicKeyInfo,
 issuerUniqueIdentifier [1] IMPLICIT UniqueIdentifier OPTIONAL,
  -- if present, version shall be v2 or v3
 subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL,
  -- if present, version shall be v2 or v3
 extensions [3] Extensions OPTIONAL
  -- If present, version shall be v3 -- }
}
```

Listing 5.2: Aufbau digitaler Zertifikate laut X.509

In Zertifikaten der X.509-Version 3 können optionale Erweiterungen eingefügt werden, die im selben Standard ebenfalls folgendermaßen formal definiert werden:

```
1 Extensions ::= SEQUENCE OF Extension
  Extension ::= SEQUENCE {
3   extnId EXTENSION.&id ({ExtensionSet}),
    critical BOOLEAN DEFAULT FALSE,
5   extnValue OCTET STRING
        -- contains a DER encoding of a value of type &ExtnType
7       -- for the extension object identified by extnId --
    }
9 ExtensionSet EXTENSION ::= { ... }
```

Listing 5.3: Aufbau von X.509-Zertifikatserweiterungen

Die Gesamtheit der X.509-Erweiterungen ist also eine Abfolge einzelner Elemente, die sich aus folgenden Angaben zusammensetzen:

- eine eindeutige Identifikationsnummer des Erweiterungstyps
- ein boolscher Wert, der festlegt, ob die Zertifikatserweiterung kritisch ist
- eine DER-kodierte [ITU02] Bytefolge, die den Nutzinhalt transportiert

Jede Extension in einem X.509-Zertifikat muss eindeutig identifizierbar sein. Das ist notwendig, damit eine Erweiterung stets korrekt verarbeitet werden kann. Hätte sie unter ein- und demselben Bezeichner bei verschiedenen Zertifikatsherausgebern verschiedene semantische Bedeutungen, wäre eine eindeutige Auswertung nicht möglich. Die Identifikation bezieht sich jedoch nicht auf eine fortlaufende Nummerierung jeder *Instanz* einer Erweiterung, sondern auf den Erweiterungstyp, der weltweit eindeutig sein soll. Für die Identifikation der Erweiterungstypen wird der hierarchische OID-Standard verwendet (gemäß ITU-T X.660, [ITU08b]). Die durch Punkte getrennten Ziffern einer OID stellen die Knotenanzahl in einem Baum dar, dessen Wurzel die Organisationen ITU-T und ISO bilden. Über Mitgliedschaften und räumliche Zuordnungen wird so eine eindeutige Zuordnung einer OID zu ihrem Inhaber möglich. Dieser erhält zudem einen eigenen Namensraum, in dem er OIDs selber vergeben kann.

Im konkreten Fall wird für die Implementation die Basis-OID des RRZN verwendet, die `1.3.6.1.4.1.18141` lautet. In Langschreibweise ist dies:

`iso.identified-organization.dod.internet.private.enterprise.RRZN`

Unterhalb dieser Basis-OID müssen Bezeichner für sämtliche Institute der Leibniz Universität Hannover untergebracht werden, weswegen eine weitere hierarchische Aufteilung im Namensraum des RRZN vorgenommen wird. So wird eine 3 für das Institut für Rechnernetze verwendet, danach eine 100 für den Forschungsbereich Grid-Computing. Nachdem diese OID bereits für die in [Pig08] vorgestellten Erweiterungen in verschiedenen Ausprägungen verwendet wurde, wird die für das Auditing benötigte mit der Unter-OID 5 bezeichnet. Die vollständige OID der Zertifikatserweiterung zum Auditing lautet somit

`1.3.6.1.4.1.18141.3.100.5.1`

Diese OID ist weltweit eindeutig. Da die Erweiterung Teil der signierten Informationen im Zertifikat ist, kann sie nicht nachträglich manipuliert werden.

5.4.2 Modifizierte Proxy-Erstellung

Im Globus Toolkit übernimmt ein Werkzeug namens `grid-proxy-init`, Teil eines Pakets namens `Proxy_Utils`, die Erstellung von Proxy-Credentials anhand des End-Entity-Zertifikats des Nutzers. Dieses Kommandozeilenwerkzeug – in C geschrieben – musste zunächst auditingfähig gemacht werden. Dazu wurde die Funktion `globus_l_gsi_proxy_sign_key()` in der Datei `globus_gsi_proxy.c` modifiziert, um eine zusätzliche X.509-Erweiterung einzufügen. Diese enthält den oder die DER-kodierte URL(s) zu einem oder mehreren Auditing-Webservices. Ein Auszug der Änderungen ist im Quellcode in Listing 5.4 zu sehen. Das Kürzel „CLMS“ steht für „Credential Lifecycle Management Service“, die in der Implementation verwendete Bezeichnung für den Auditingdienst.

```
1  if(handle->clms_url)
   {
3    ASN1_STRING *       clms_DER_string;
     int                 clms_DER_length;
5    char *              clms_url = NULL;
     int                 clms_NID;
7    unsigned char *     clms_DER;
     ASN1_OBJECT       * obj;
9
     clms_url = malloc(strlen(handle->clms_url));
11   clms_url = handle->clms_url;
     clms_DER_length = strlen(clms_url);
```

```
13    clms_DER = malloc(clms_DER_length);
      clms_DER_string = ASN1_STRING_new();
15    ASN1_STRING_set(clms_DER_string, clms_url, clms_DER_length);
      clms_NID = OBJ_create(  "1.3.6.1.4.1.18141.3.100.5.1",
17                            "CLMS-URL",
                              "Credential Lifecycle Management Service -
                                 URL");
19    obj = OBJ_nid2obj(clms_NID);
      extension = X509_EXTENSION_create_by_OBJ(NULL,
21                                             obj,
                                               0,
23                                             clms_DER_string);
      ASN1_STRING_free(clms_DER_string);
25    X509_add_ext(*signed_cert, extension, 0);
      X509_EXTENSION_free(extension);
27  }
```

Listing 5.4: Auszug aus modifizierter globus_gsi_proxy.c

Da der Einbau der Zertifikatserweiterung vom Nutzer bestimmt werden soll, wurde eine zusätzliche Option für das Kommandozeilenwerkzeug `grid-proxy-init` eingeführt, die `-clms` lautet und eine URL erwartet. Mit dem beispielhaften Aufruf:

```
grid-proxy-init \\
-clms https://audit.grid.uni-hannover.de:8443/Services/CLMService
```

kann der Grid-Nutzer nunmehr ein Proxy-Credential erzeugen, das von entsprechend erweiterten Grid-Ressourcen auditiert wird. Die Kompatibilität der Proxy-Credentials mit nicht auditingfähigen Ressourcen wird dabei nicht berührt. Alle Credentials sind – soweit in `grid-proxy-init` vorgesehen – RFC3820-kompatibel und besitzen somit auch eindeutige Seriennummern, die vom Auditingdienst zur Identifikation benötigt werden.

5.5 Auditing-Webservice

Ein wichtiger Teil der Proof-of-Concept-Implementation ist der zentrale Webservice, der Auditinginformationen von den, gemäß Abschnitt 5.3 modifizierten, Grid-Ressourcen entgegennimmt. Dieser Webservice stellt zunächst mittels einer über das

WSRF-Protokoll standardisierten Schnittstelle den Endpunkt für die Kommunikation mit auditingfähigen Grid-Ressourcen bereit, nimmt Auditingdaten entgegen und speichert sie mittels eines Persistenzlayers.

Der Webservice wurde in Java geschrieben und basiert – wie in der Entwurfsphase (vgl. Abs. 4.4.4) festgelegt – auf dem Globus Container. Er wurde mithilfe des MAGE GDT [FSF06] implementiert, das alle Grundfunktionen inklusive der WSRF-Protokollabstraktion zur Verfügung stellte. Auch die „Stubs“, also Grundfunktionen für eine Client-Anwendung, wurden durch das GDT bereitgestellt. In einer annotierten Hauptklasse werden Methoden, die später per WSRF aufgerufen werden sollen – also insbesondere jene zur Speicherung eines neuen Auditing-Eintrags – mit `@GridMethod` markiert. Die durch das GDT erzeugte Klasse `CLMService` beinhaltet alle für den Webservice notwendigen Methoden.

Nach der Annahme eines Auditing-Records durch den Webservice wird dieser in einer Instanz der Klasse `AuditRecord` gespeichert und mittels Data Access Objects an die Datenbank übermittelt.

Der Globus Container und die mittels GDT generierten Grundfunktionen kapseln bereits alle relevanten Sicherheitsmechanismen, so dass die Beispielimplementation ohne zusätzlichen Implementationsaufwand über volle GSI-Unterstützung verfügt. Kommunikation mit dem Auditing-Dienst ist – sofern der ihn beherbergende Globus Container entsprechend konfiguriert ist – nur über einen GSI-geschützten Kanal möglich; die Authentifizierung mittels X.509-Proxy-Zertifikaten und eine gridmapfile-basierte Autorisierung sind ebenfalls implementiert.

5.5.1 Datenstruktur für Auditing-Einträge

Für die Auditingeinträge wurde eine passende Datenstruktur in Form einer Java-Klasse namens `AuditRecord` implementiert. Diese orientiert sich an den in Abs. 4.4.3 genannten Daten und speichert diese wie folgt:

- `serialNr`: die Seriennummer des zu auditierenden Proxy-Zertifikats liegt als Ganzzahl vor
- `issuer`: die Seriennummer desjenigen Zertifikats (Proxy- oder Nutzerzertifikat), das zur Ableitung des zu auditierenden Zertifikats verwendet wurde
- `dn`: der Distinguished Name des zu auditierenden Zertifikats

- `timestamp`: Zeitstempel des Auditing-Eintrags, Auflösung 1ms
- `host`: Vollqualifizierter Hostname (FQDN) des Hosts, der den Auditing-Record versandt hat
- `service`: wenn programmatisch ermittelbar (s.u.), der Name des Dienstes, der das auditierte Credential genutzt hat
- `usageMode`: Art der Nutzung des Credentials (Authentifizierung oder Delegation)

Während die Herkunft und Bedeutung der meisten dieser Variablen offensichtlich ist, bedürfen einige näherer Erläuterung. Bei der in `usageMode` gespeicherten Art der Nutzung wird zwischen der Nutzung eines Proxy-Credentials zur Authentifizierung (also etwa bei einem GSI-gestütztem Webserviceaufruf) und zur Delegation (also einer Ableitung eines neuen Proxy-Zertifikats für einen Grid-Job) unterschieden. Der Dienstname (Variable `service`) wird von der Klasse `AuditRecorder` eingesetzt, wenn diese den aufgerufenen Webservice ermitteln konnte (siehe Abschnitt 5.2).

5.5.2 Persistenz

Alle eingehenden validen Datensätze werden vom Auditing-Webservice persistent gespeichert; diese Funktionalität ist essenziell für das korrekte Funktionieren der Auditing-Infrastruktur. Für die Speicherung und den Zugriff auf Auditingdaten durch den Java-Webservice wurde ein Data Access Object (DAO) entwickelt, das einige Methoden zum Verbindungsmanagement, zur Datenein- und ausgabe kapselt. Diese Schnittstelle wird sowohl in Java als auch in PHP – der Sprache, in der das in Abs. 5.6 vorgestellte prototypische Webinterface entwickelt wurde – zur Verfügung gestellt und bietet eine möglichst hohe Flexibilität und Erweiterbarkeit.

In der Implementierung ist keine Methode zur Löschung oder Veränderung einzelner Datensätze vorgesehen. Auditing-Daten sind nach der Speicherung nicht mehr veränderbar. Diese Einschränkung ist konzeptionell beabsichtigt, damit Angriffe gegen die Auditing-Protokolle nicht zu einer Korrumpierung des Datenbestandes führen können.

Daten können in der aktuellen Implementierung entweder in einer relationalen MySQL-Datenbank oder als XML-Datei im Dateisystem gespeichert werden. Die

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
serialNr	varchar(50)	NO		NULL	
issuer	varchar(50)	NO		NULL	
timestamp	varchar(30)	NO		NULL	
dn	varchar(200)	NO		NULL	
host	varchar(200)	YES		NULL	
service	varchar(100)	NO		NULL	
cred_usage	varchar(50)	NO		NULL	

Tabelle 5.2: Schema der Auditing-Tabelle

Schnittstelle zur MySQL-Datenbank erfolgt über JDBC und den Treiber MySQL-Connector/J; andere RDBMS können mit geringem Entwicklungsaufwand angebunden werden. Die Speicherung und das Auslesen der Datensätze erfolgt mittels Prepared Statements; mögliche Angriffe über SQL-Injection (etwa durch speziell präparierte Auditing-Einträge) werden somit wirksam unterbunden.

Das verwendete RDBMS MySQL liegt als Open-Source-Produkt vor und wird in der unten beschriebenen Testumgebung auf demselben virtuellen Server betrieben, auf dem auch der Auditing-Webservice beheimatet ist. Der Datenbankserver muss also nicht über eine Netzwerkverbindung angesprochen werden – so werden Angriffe gegen das Datenbanksystem de facto unmöglich. Da die Konfiguration der Datenbankanbindung ohne Anpassungen im Quellcode mittels einer Textdatei vorgenommen werden kann, ist jedoch auch hier maximale Flexibilität gegeben.

Zusätzlich zur Datenspeicherung in einem relationalen Datenbanksystem wurde die Möglichkeit implementiert, Auditing-Datensätze lokal im Dateisystem abzulegen. Diese Möglichkeit ist jedoch primär zu Entwicklungs- oder Portierungszwecke vorgesehen und verwendet JDOM, um Auditing-Records in einer XML-Datenstruktur zu speichern. Die gespeicherten Nutzdaten sind bei beiden Speicherarten identisch.

5.6 Prototypisches Webinterface

Um Nutzern eine Möglichkeit zu geben, die in ihrem Namen gsammelten Auditingdaten bequem einzusehen, wurde ein prototypisches webbasiertes Interface (siehe Abb. 5.2) implementiert. Dieses Webinterface stellt alle gesammelten Auditingdaten in einer übersichtlichen HTML-Ansicht dar und erlaubt somit einen schnellen Überblick, ob und welche Auditing-Informationen vorhanden sind. Es wurde mithilfe

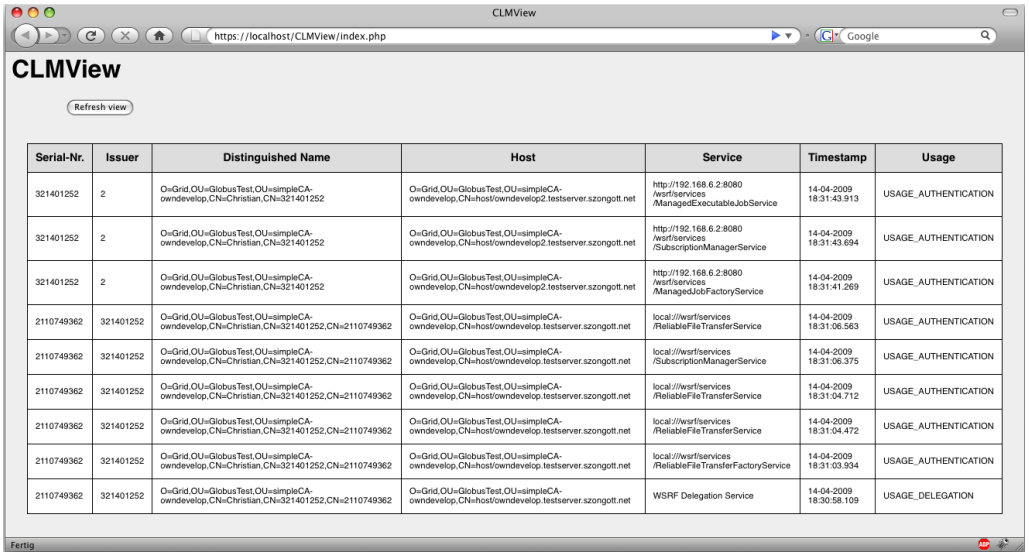


Abbildung 5.2: Webbasierte Oberfläche zur Analyse von Auditingdaten

der Skriptsprache PHP entwickelt und greift über dieselbe Datenbank-Abstraktion auf die Auditing-Datenbank zu wie der Webservice. In der Testbed-Umgebung läuft es ebenfalls auf demselben virtuellen Server, kann jedoch auch auf einen externen Rechner ausgelagert werden (unter Berücksichtigung der damit einhergehenden Sicherheitsimplikationen). Da das Webfrontend nicht im Globus Container läuft, wird ein separater Webserver benötigt; in der Testbed-Umgebung wurde Apache verwendet.

5.6.1 Sicherheit des Webinterface

Bei der Implementierung einer webbasierten Software sind verschiedene Sicherheitsaspekte zu betrachten; diese sind sowohl programmatischer [KE08] als auch konzeptioneller Natur. Auf programmatischer Ebene muss die entwickelte Software frei von Fehlern sein, die eine Privilegien-Eskalation oder unerwünschten Zugriff auf Informa-

tionen ermöglichen; dem prototypischen Webinterface können beide Eigenschaften zugesprochen werden.

Wichtiger ist jedoch eine adäquate Vertraulichkeit, Authentifizierung und Autorisierung beim Zugriff auf die sensitiven Auditingdaten. Nutzer der Auditing-Infrastruktur sollen schließlich nur jene Daten einsehen dürfen, die sie selber betreffen; ein Abfangen von Daten darf nicht möglich sein. Die Vertraulichkeit der Daten wird dadurch gewährleistet, dass der Apache-Webserver eingehende Verbindungen nur über SSL entgegennimmt; das dazu notwendige Modul `mod_ssl` befindet sich im Lieferumfang des Webservers.

Auch die Skriptsprache PHP verfügt über SSL-Unterstützung; somit kann die Authentifizierung des Nutzers ebenfalls über SSL erfolgen. Ist das Grid-Zertifikat des Nutzers in seinem Browser gespeichert (bei Nutzern des D-Grid ist dies regelmäßig der Fall, da Zertifikate mittels eines webbasierten Workflows beantragt werden), können Browser und Webserver eine gegenseitige Authentifizierung durchführen und die Metadaten des vorgezeigten Nutzerzertifikats wird als Umgebungsvariable an die ausgeführten PHP-Skripte übertragen. Nun kann zunächst geprüft werden, ob ein gültiges und akzeptiertes Zertifikat vorliegt; der DN wird sodann als Filter an das Datenbanksystem übergeben, um nur jene Datensätze anzufragen, die das vorgezeigte Nutzerzertifikat direkt betreffen.

Mittels dieses einfachen Mechanismus kann sichergestellt werden, dass nur jene Nutzer Zugriff auf die Auditingdaten erhalten, die auch hierfür autorisiert sind. Anfragen durch Nutzer, die nicht über ein akzeptiertes Zertifikat verfügen, werden abgelehnt.

5.7 Überwachung der Auditierung

In Abschnitt 4.5 wurde skizziert, dass Angriffe gegen die Auditing-Infrastruktur an verschiedenen Punkten ansetzen können und durch das Auditing-System unter Vermeidung inkompatibler Änderungen abgewehrt werden sollen.

Zu diesem Zweck ist eine zusätzliche Komponente konzipiert und in der Veröffentlichung [KWS10] erstmals vorgestellt worden. Diese zusätzliche Komponente, der „Audit Watchdog“, agiert innerhalb einer Grid-Site als Prüfinstanz und gleicht in Zusammenarbeit mit dem Auditing-Dienst die an diesen gelieferten Datensätze mit einer Liste der innerhalb der Grid-Site vorhandenen auditingfähigen Dienste ab.

Die neue Komponente agiert als Netzwerk-Sniffer und überwacht den verschlüssel-

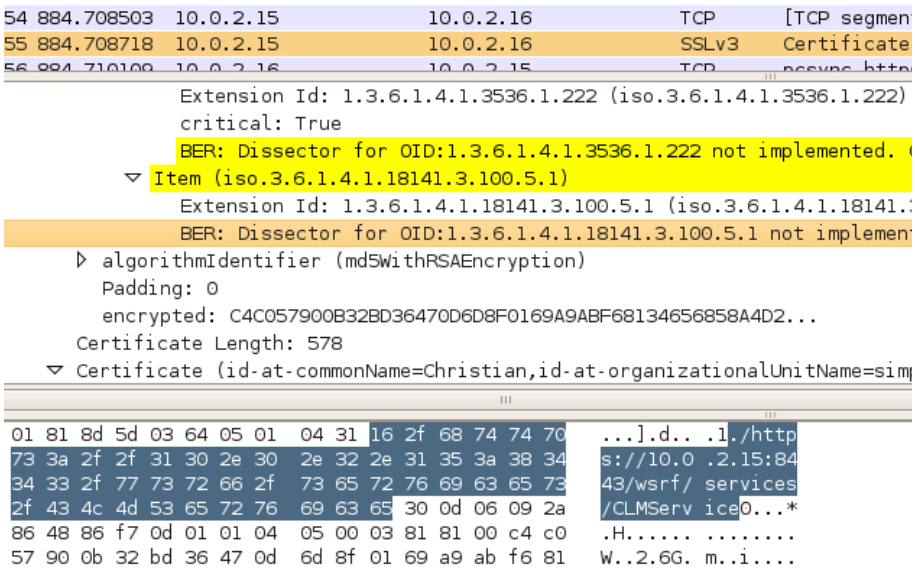


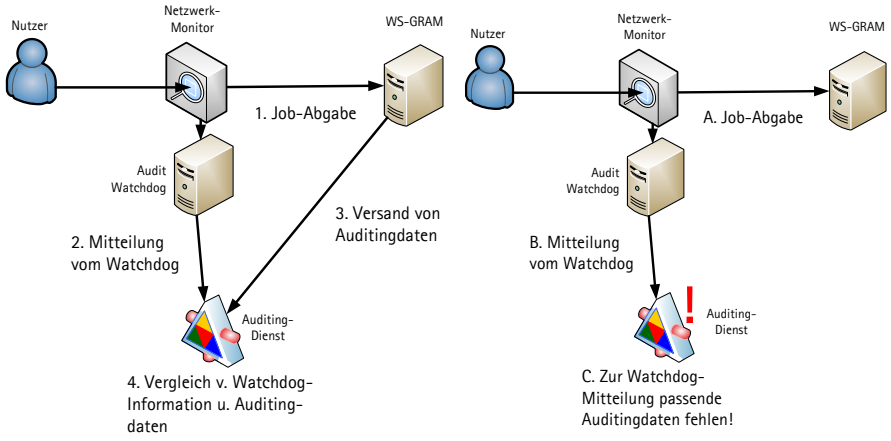
Abbildung 5.3: Entschlüsseltes SSL-Paket mit Auditing-Informationen

ten Datenverkehr zwischen verschiedenen Ressourcen in einer Grid-Site. Durch die Plazierung in demselben Switch-Segment (bzw. in einer ähnlichen netztopologisch zentralen Position im LAN der Grid-Site) hat diese Komponente die Möglichkeit, jeglichen Datenverkehr zwischen den Ressourcen der Grid-Site abzuhören und – sofern ihr die Hostzertifikate der Grid-Sites nebst privatem Schlüssel zur Verfügung stehen – zu entschlüsseln (siehe Abb. 5.3). Sobald während einer Kommunikation ein Proxy-Zertifikat mit Auditing-Erweiterung detektiert wird, sendet die Watchdog-Komponente eine Kontrollnachricht an den Auditing-Dienst. Folgt diesem nicht in einem bestimmten Zeitraum der entsprechende Auditing-Record, so muss davon ausgegangen werden, dass ein Problem vorliegt. Die Abläufe bei ordnungsgemäßem und defektem Auditing sind in Abbildung 5.4 illustriert. Der ordnungsgemäße Ablauf des Auditing mithilfe eines Audit-Watchdogs ist in Abb. 5.4a skizziert. Ist das Auditing auf der mit „WS-GRAM“ beschrifteten Grid-Ressource aktiviert, sind folgende Schritte zu verzeichnen:

1. Der Nutzer sendet einen Grid-Job an den auditingfähigen WS-GRAM.

2. Die Auditing-Erweiterung im Proxyzertifikat wird vom Audit-Watchdog erkannt und der Auditing-Dienst wird benachrichtigt.

3. Der WS-GRAM sendet einen Auditing-Record an den Auditing-Dienst.



(a) Ordnungsgemäßes Auditing (b) Auditing gestört

Abbildung 5.4: Audit Watchdog im Zusammenspiel mit Auditing-Infrastruktur

4. Der Auditing-Dienst korreliert die beiden empfangenen Nachrichten und stellt eine Übereinstimmung fest.

Fehlt hingegen der Auditing-Record, wie in 5.4b dargestellt, so kann die im letzten Schritt durch den Auditing-Dienst vorgenommene Korrelation mangels Eingabedaten nicht erfolgreich sein. Stellt der Auditing-Dienst zudem fest, dass die betreffende Grid-Ressource früher bereits erfolgreich auditiert wurde, so kann das Fehlen des Audit-Records nur zwei Ursachen haben:

1. Die betreffende Komponente ist generell nicht auditingfähig oder

2. Ihre Auditing-Fähigkeit wurde durch einen Angriff oder eine Fehlkonfiguration außer Kraft gesetzt.

Der zweite Fall deutet auf einen möglichen Missbrauch hin und sollte zu einer Warnung an den Nutzer und ggf. an den Administrator der betroffenen Grid-Komponente führen. Durch die zusätzlichen Informationen, die der Audit Watchdog anhand der direkt im Netzwerk gesammelten Daten liefert, kann ein vom Angreifer manipuliertes Clientsystem, das entweder nicht mit dem Auditing-Service kommunizieren kann oder aufgrund einer Änderung der GSI-Bibliotheken nicht mehr auditingfähig ist, entlarvt werden. Somit schließen sich zwei der in den vorigen Abschnitten aufgeworfenen Angriffsvektoren. Tabelle 5.3 zeigt die veränderte Bedrohungssituation, nachdem der Auditing-Watchdog als zusätzliche Warnkomponente integriert wurde.

Angriff	Angriff abwehrbar	m. Watchdog abwehrbar
Entfernen der Zertifikatserweiterung	✓	✓
Störung des Auditing-Dienstes	✓	✓
Unterbinden der Kommunikation zwischen Host und Auditing-Dienst[1]		✓
Manipulation der GSI-Bibliotheken[1]		✓

Tabelle 5.3: Mit Watchdog abwehrbare Angriffe gegen die Auditing-Infrastruktur

5.7.1 Fazit

Angriffe gegen die Auditing-Infrastruktur, die auf „Denial of Service“ oder einer Manipulation der eingesetzten Proxyzertifikate basieren, sind vom Auditing-System leicht abzuwehren. Wenn ein Angreifer jedoch ein Clientsystem übernommen und mittels seiner administrativen Privilegien das Auditing deaktiviert hat, kann das Auditing-System diesen Angriff nicht feststellen. Mithilfe einer optionalen, zusätzlichen Komponente ist es möglich, solche ansonsten schwer detektierbaren Angriffe, die die Kommunikation zwischen Host und Auditing-Dienst administrativ unterbinden oder direkt die GSI-Bibliotheken austauschen, zu erkennen.

Somit sind alle denkbaren Angriffe gegen die Auditing-Infrastruktur durch diese abwehrbar.

5.8 Auditing-Testbed

Zur Erprobung der Implementation wurde diese in einem virtualisierten „Testbed“, also einer in sich geschlossenen Testumgebung, installiert und konfiguriert. Dieses Testbed bestand aus drei virtuellen Linux-Servern, auf denen jeweils das Globus Toolkit 4.0.8 installiert wurde. Auf jeder der virtuellen Maschinen lief der Globus Container; es wurden neben den Standarddiensten (wie etwa `AuthenticationService`

[1] Administrative Privilegien auf dem angegriffenen Server notwendig

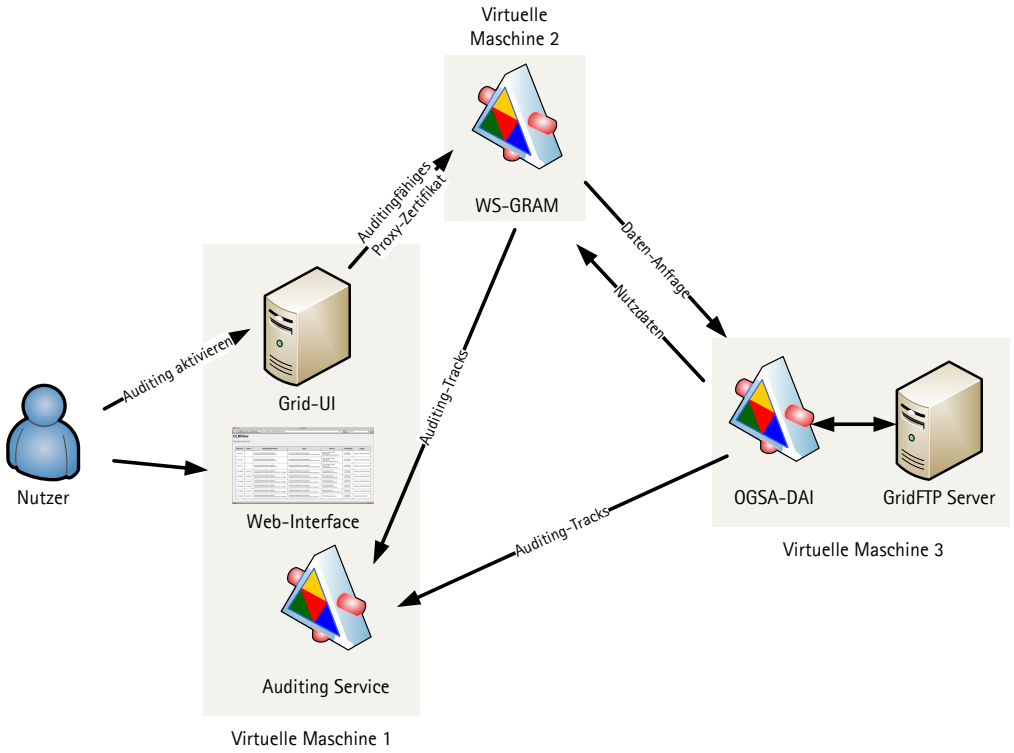


Abbildung 5.5: Aufbau der Testbed-Installation

und `DelegationService`) jedoch unterschiedliche Services deployed:

- Ein WS-GRAM nebst Ausführungsmöglichkeit einfacher Jobs diente als Testumgebung für die Abgabe von Grid-Jobs.
- Auf einer virtuellen Maschine wurde der Auditing-Webservice nebst dem webbasierten Nutzerinterface eingespielt; dieser Server diente auch als UI-Host für die Abgabe von Jobs.
- Ein dritter Server mit dem OGSA-DAI-Framework wurde zum Test des Auditings von Datenzugriffen genutzt.

Neben der Funktionsüberprüfung des Auditings war die Testbed-Installation insbesondere dafür konzipiert, unter möglichst realistischen Bedingungen Nutzdaten für die in Kapitel 6 beschriebene Angriffserkennung zu sammeln; hier wurden automatisiert mehrere Tausend Jobs submittiert, um eine ausreichende Datenbasis zu erhalten. Auch einige Szenarien für Missbrauchsfälle konnten im Testbed nachvollzogen werden; da dieses anders als die produktive Grid-Infrastruktur komplett unter der Kontrolle der Tester stand, waren keine Schäden zu befürchten.

5.9 Fazit

Mit der prototypischen Implementation der Auditing-Infrastruktur auf Basis des Globus Toolkit 4.0.8 wurde die Praxistauglichkeit des Konzepts unter Beweis gestellt. Unter Beachtung aller für die Entwicklung von Grid-Diensten relevanten Standards wurde zunächst die Grid Security Infrastructure modifiziert, um eine X.509-Erweiterung in Globus-Proxy-Zertifikate zu integrieren, die diese als auditingfähig kennzeichnet. Die GSI wurde in einem zweiten Schritt angepasst, damit Auditing-Informationen während der sicherheitsrelevanten Authentifizierungs- und Delegationsvorgänge gesammelt werden konnten. Im Rahmen des „Proof of Concept“ wurde zudem ein WSRF-Webservice erstellt, der – in einem Globus-Container laufend – Auditingeinträge über eine sichere Schnittstelle entgegennimmt und weiterverarbeitet.

Bei der Implementation der Auditing-Infrastruktur konnten wertvolle Erfahrungen gesammelt und der Grundstock für eine weitere Entwicklung etwa im Rahmen des „Globus Incubator“-Projekts (siehe Abs. 8.2.2) gelegt werden.

Kapitel 6

Angriffserkennung mit Bayes-Klassifikatoren

Die durch eine Auditing-Umgebung gesammelten Informationen sind schon aufgrund ihrer schieren Menge für den Grid-Nutzer wenig aussagekräftig. Ein Erkennungsmechanismus soll anhand verschiedener Kriterien der Audit Trails feststellen, ob eine missbräuchliche Nutzung vorliegt.

Im folgenden Kapitel wird zunächst diskutiert, welche Erkennungsmechanismen für das Grid-Auditing geeignet sind. Anhand einer umfassenden Simulation wird sodann die Umsetzbarkeit automatisierter Möglichkeiten der Missbrauchserkennung nachgewiesen.

Die in diesem Kapitel diskutierten Arbeiten zur Simulation und Missbrauchserkennung wurden in Kooperation mit Nina Tahmasehbi vom Forschungsinstitut L3S durchgeführt und in einem Paper auf der Konferenz Grid2011 veröffentlicht [KTRS11].

6.1 Motivation

Die Daten, die vom in den vorigen Kapiteln dieser Arbeit vorgestellten System gesammelt werden, stehen in Form sogenannter „Audit Trails“ zur Verfügung. Wie in Abschnitt 2.4.4 erläutert, handelt es sich bei den Audit Trails bereits um eine Aggregation der erfaßten Rohdaten. Diese werden insofern zusammengefasst, als dass alle Zertifikatsdaten, die demselben ursprünglichen Proxy-Credential abstammen, also dessen Weg durch das Grid darstellen, anhand ihrer Seriennummern zu einem Datensatz zusammengeführt werden. Jeder Audit Trail stellt also den Pfad eines Proxy-Zertifikats und seiner Nachkommen durch das Grid dar.

Eine weitere Aggregation der Rohdaten ergibt sich naturgemäß daraus, dass ein Grid-Nutzer nur die ihn betreffenden Auditing-Daten einsehen und für eine Missbrauchserkennung nutzen darf; dementsprechend werden die Daten nur dann angezeigt, wenn der Inhaber des ursprünglichen End-Entity-Credential mit dem des ausstellenden Zertifikats der Proxy-Credentials übereinstimmt.

Über diese recht grobgranularen Methoden der Datenaggregation hinaus wird durch den Auditing-Dienst jedoch keine weitere Verarbeitung der Rohdaten vorgenommen. Für den Nutzer des Auditing-Service ergibt sich nun die Schwierigkeit, aus der Fülle der ihm zur Verfügung stehenden Audit Trails – welche wiederum durch Mehrfach- und Parallelableitungen recht verzweigt sein können – zu ermitteln, ob die von ihm an die Grid-Ressourcen delegierten Credentials missbraucht wurden oder nicht. Es ist eine der Grundideen des Grid-Computing, dass Rechenjobs auf verschiedensten Ressourcen ausgeführt und Daten ebenso diversifiziert gespeichert werden. Daher kann nicht vorausgesetzt werden, dass der Nutzer die Rohinformation über Zeitpunkt und Ort einer Authentifizierung/Delegation korrekt deuten kann.

Um einen echten Sicherheitsgewinn für den Nutzer zu bieten, ist es daher notwendig, diesen bei der Erkennung, ob seine Grid-Credentials missbräuchlich genutzt wurden, zu unterstützen. Diese Unterstützung muss automatisiert erfolgen und sollte so wenig wie möglich in den Arbeitsablauf des Nutzers eingreifen. Er soll vielmehr beim Verdacht auf Missbrauch auf einen Blick erkennen können, ob das Auditing-System einen Verdachtsfall als missbräuchlich einstuft oder nicht; eine zuverlässige Ja-Nein-Entscheidung ist also gefragt.

6.2 Methoden zur automatisierten Missbrauchserkennung

Nutzer aus verschiedenen wissenschaftlichen Communities vergeben sehr unterschiedliche Aufgaben an eine Grid-Infrastruktur und ihre Auditing-Profile werden sich demnach voneinander unterscheiden. So nutzen Astrophysiker das Grid, um mittels einer Spezialsoftware anhand von Gravitationswellen bislang unentdeckte Himmelskörper zu ermitteln – eine Aufgabe, die zwar praktisch unbegrenzte Rechenleistung, jedoch nur vergleichsweise wenig persistente Speicherkapazität auf den Compute-Nodes oder externen Storage-Servern benötigt. Andere Wissenschaftler nutzen das Grid, um die immensen Datenmengen des Large Hadron Collider (LHC) zu analysieren – eine Aufgabe, die sowohl Rechen- als auch Speicherkapazität benötigt und durch regelmäßige hochvolumige Datentransfers zwischen verschiedenen Grid-Ressourcen gekennzeichnet ist.

Eine automatische Missbrauchserkennung muss diese Fälle berücksichtigen; ihr stehen dabei nur recht wenige Informationen über die Nutzer – nämlich praktisch ausschließlich die Daten, die im Auditing-Trail enthalten sind – zur Verfügung.

Bei der Spam- und Virenbekämpfung, bei Intrusion-Detection-Systemen oder in Firewalls werden im Wesentlichen zwei verschiedene Methoden der Entscheidungsfindung verwendet:

1. **Regelbasierte Systeme** wie etwa Firewall-Regeln oder die Wortfilter-Regeln eines Spamfilters.

2. **Heuristische oder wahrscheinlichkeitsbasierte Methoden** wie etwa bei Virenscannern oder wahrscheinlichkeitsbasierten Spamfiltern,

Beide Methoden haben distinkte Vor- und Nachteile und eignen sich in verschiedenen Anwendungsfällen unterschiedlich gut.

6.2.1 Regelbasierte Systeme

Die Entscheidungsfindung auf Basis eines festen Regelsatzes ist ein sehr verbreiteter Ansatz in praktisch jeder Wissenschaft, auch der Informatik. Da diese Form der Entscheidungsfindung dem intuitiven Vorgehen des Menschen bei komplexen Zusammenhängen sehr nah ist, können regelbasierte Entscheidungen leicht nachvollzogen werden. Sie eignen sich jedoch nur für stets wiederkehrende Situationen,

deren Dynamik durch das Regelwerk beherrschbar ist. So ist die Verwendung von festen Regelwerken bei Firewalls noch immer das Standardvorgehen; hier wird für jede zu schützende Port/IP-Adress-Kombination eine Regel definiert, auf deren Basis das Firewallsystem über Annahme oder Ablehnung eines Datenstroms/-pakets entscheidet. Häufig handelt es sich dabei um eine Whitelist, es sind also nur Regeln für zu akzeptierende Datenströme definiert. Alle anderen Kommunikationsversuche werden unterbunden.

Intrusion-Detection-Systeme fällen eine Vielzahl von Entscheidungen ebenfalls auf der Basis eines Regelwerks, nämlich einer Datenbank mit Signaturen bekannter Angriffe. Trifft keine der Regeln auf die untersuchte Kommunikation zu, so handelt es sich nach dem Kenntnisstand des IDS nicht um einen Angriff – das Regelwerk ist hier in Form einer Blacklist implementiert.

Im Grid-Computing wird vielfach auf Regelsysteme zur Autorisierung zurückgegriffen, etwa im Rahmen des CAS (Community Authorization Service) [PWF+02] oder dem im Globus Toolkit 4 enthaltenen „Multipolicy Authorization Framework“ [LFS+06]. Diese Regelwerke treffen eine binäre Entscheidung, ob ein Zugriff gestattet oder abgelehnt werden soll. Auch hier gilt jedoch: Autorisierungsvorgänge sind wenig dynamisch; die Regelwerke müssen selten an eine neue Situation angepaßt werden.

Der Hauptnachteil regelbasierter Systeme ist die mangelnde Flexibilität, die sich vor allem darin äußert, dass das Regelwerk – das in der Regel von einem Experten vor Benutzung des Systems erstellt wird – nicht in einer ausreichend kurzen Zeitspanne an sich ändernde Gegebenheiten angepasst werden kann; eine Feedbackschleife fehlt.

6.2.2 Statistische und heuristische Verfahren

In vielen Feldern der computergestützten Entscheidungsfindung haben sich in den letzten Jahren Verfahren etabliert, die auf wahrscheinlichkeitsbasierten Modellen basieren oder sich heuristischer[1] Ansätze bedienen. Der bei der Nutzung dieser Verfahren eingegangene Tradeoff erlaubt Abstriche bei der Entscheidungsgenauigkeit zugunsten eines insgesamt günstigeren Laufzeitverhaltens und ist somit insbesondere in Umgebungen mit hoher Problemfrequenz sinnvoll.

Heuristische Verfahren finden sich in der Informatik etwa in der Graphentheorie.

[1] Heuristik (gr. *heuriskein*, finden, entdecken) bezeichnet Verfahren, die mit relativ geringem (Rechen-)Aufwand zu guten, wenn auch nicht optimalen Lösungen kommen.

In der praktischen Anwendung im Sicherheitsbereich sind Heuristiken insbesondere in der Malware-Bekämpfung, also bei Virenscannern und Anti-Rootkit-Werkzeugen, im Einsatz. Hier werden allgemeine Verhaltensmerkmale eines Programm als Indikatoren für seine mögliche Bösartigkeit verwendet, was im Einzelfall zu spektakulären Mißerfolgen führt (wenn etwa systemkritische Komponenten fälschlicherweise als bösartig eingestuft werden), in der Regel aber gute Ergebnisse liefert.

Ein weiteres Anwendungsbeispiel für heuristische Methodik ist die automatische Klassifizierung von Texten mithilfe Bayesscher Netzwerke und Klassifikatoren. Diese Anwendung, eingeführt in einem Paper des Autors Paul Graham [Gra02], hat sich als geeignete Methode zur Einordnung von E-Mails als Spam durchgesetzt und ist mittlerweile in vielen entsprechenden Programmen enthalten. Basierend auf einer möglichst großen Menge an Trainingsdaten (für Spam und Nicht-Spam) errechnet der Algorithmus zunächst die Worthäufigkeit der enthaltenen Wörter, um darauf aufbauend dann die Wahrscheinlichkeit zu errechnen, mit der ein einzelnes Wort in einer Spam-Mail vorkommt.

Wichtige Maßzahl jedes Verfahrens ist die Zahl der falschen Entscheidungen. Hier wird zunächst differenziert: Ein False Positive liegt vor, wenn ein Kriterium vom zu prüfenden Datensatz nicht erfüllt wird, der Algorithmus es jedoch als erfüllt betrachtet; False Negatives hingegen bezeichnen ein vom Datensatz erfülltes Kriterium, das fälschlich nicht erkannt wird. Eine zu Unrecht als Virus identifizierte Systemdatei wäre demnach als False Positive zu werten; eine durch den Klassifizierungs-Algorithmus nicht gefundene Spam-Mail hingegen ist ein False Negative.

6.3 Bewertung der Entscheidungsverfahren

Bei der Entscheidung für ein Verfahren zur automatischen Erkennung des Missbrauchs von Proxy-Credentials spielten mehrere Faktoren eine Rolle. In der Reihenfolge ihrer Wichtigkeit für das Auditing-System lauten wichtige Kriterien wie folgt:

A Flexibilität und Adaption des Algorithmus an verschiedene Nutzerprofile

B Erkennungsgenauigkeit; insbesondere geringe Falsch-Negativ-Rate

C Geschwindigkeit / Zeitkomplexität

Im (idealen) Falle einfacher, nichtrekursiver Regeln würde sich die *Zeitkomplexität* eines Regelwerks im Bereich von $\mathcal{O}(n)$ für eine Anzahl von n Regeln bewegen; bei verketteten oder rekursiven Regeln jedoch verschlechtert sich dieses Laufzeitverhalten. Es ist nicht zu erwarten, dass für die Klassifikation der Nutzung von Proxy-Credentials einfache Regelwerke ohne Abhängigkeiten der Regeln untereinander genügen.

Es ist jedoch anzumerken, dass die Zeitkomplexität und daraus resultierend die Geschwindigkeit des Erkennungsalgorithmus von nachrangiger Bedeutung ist, da die Missbrauchserkennung nicht während der Kommunikation, sondern asynchron nach Eingang vollständiger Audit Trails beim Auditing-Dienst durchgeführt wird. Die Kommunikation im und mit dem Grid – die von den Entwicklern der Grid-Middleware für hohen Durchsatz ausgelegt ist – wird daher durch den Detektionsalgorithmus nicht beeinflußt.

Ein weiteres Kriterium ist die zu erwartende *Erkennungsgenauigkeit* und Rate falsch positiv/negativ erkannter Merkmale. Diese Maßzahlen a priori abzuschätzen, ist schwierig, hängen sie doch sehr von den Umständen der Implementation ab. Wird ein Detektionsalgorithmus auf Basis eines unvollständigen Regelwerks realisiert, so wird sich dies erheblich auf die Genauigkeit der Erkennung auswirken. Bei wahrscheinlichkeitsbasierten und heuristischen Ansätzen wirken sich dementsprechend Faktoren wie die Anzahl der verfügbaren Merkmale und die Größe des Trainingssets für Positiv- und Negativmerkmale auf das Endergebnis deutlich aus.

Im vorliegenden Fall stehen relativ wenige Merkmale für wahrscheinlichkeitsbasierte Analyse zur Verfügung (nämlich der Ableitungspfad eines Proxy-Credentials, sein Besitzer, Gültigkeitsdauer, Verwendungszweck und -ort und Seriennummer). Daher ist zu erwarten, dass die Rate der False Positives oder False Negatives vergleichsweise hoch ist. Schwerer wiegt jedoch der Umstand, dass die Aufstellung eines erschöpfenden Regelwerks für die Grid-Nutzung unmöglich erscheint (s.u. in Abschnitt 6.3.1).

Das aus Sicht des Autors wichtigste Kriterium bei der Wahl eines Algorithmus zur Entscheidungsfindung ist jedoch die Adaption des Algorithmus an sich ändernde Situationen. Da im Grid Computing eine Vielzahl von Ressourcen in einem teilweise hochdynamischen Ressourcenverbund (modelliert durch virtuelle Organisationen, siehe Abs. 2.2.1) organisiert sind, deren Nutzerschaft sich zudem häufig ändert, muss ein Algorithmus, der Missbrauch in einem solchen Verbund erkennen soll, hochagil

sein. Stießen neue Nutzer aus einem bislang nicht vertretenen Fachgebiet zu einer virtuellen Organisation hinzu, müßte ihr Nutzungsprofil bei einem regelbasierten Ansatz zunächst vom Autor des Regelwerks aufgezeichnet und analysiert werden, um auf dieser Analyse basierend neue Regeln zu schreiben, die auf das hinzugekommene Nutzungsprofil passen. Im Gegenzug müßte geprüft werden, ob die neuen Regeln zu unerwünschten Seiteneffekten bei Analyse des Verhaltens der bestehenden Nutzerschaft führen. Der resultierende Aufwand für die Pflege eines Regelwerks scheint nicht wünschenswert. Wahrscheinlichkeitsbasierte Algorithmen hingegen benötigen zwar ebenfalls eine Aufzeichnung des Nutzerverhaltens – also ein neues Set von Trainingsdaten –, eine manuelle Neuformulierung der Wahrscheinlichkeiten ist jedoch nicht notwendig, weil das Modell sich basierend auf den erweiterten Trainingsdaten neu aufbaut.

6.3.1 Aufstellung eines Regelwerks

Es ist nach derzeitigem Kenntnisstand zwar davon auszugehen, dass Mitglieder derselben virtuellen Organisation – in ihrem grundsätzlichen Nutzungsverhalten einander ähneln, jedoch ist diese Ähnlichkeit zum Einen nicht messbar (da keine Aufzeichnungen darüber im großen Stil existieren), zum Anderen sind bei einigen analysierten Beispielcommunities auch innerhalb einer VO merkbare Unterschiede im Nutzungsverhalten zu beobachten gewesen. Desweiteren ist anzunehmen, dass auch die Grid-Nutzung ein- und desselben Anwenders variieren wird. Dies kann leicht an einigen Beispielen veranschaulicht werden:

- Ein neuer Grid-Job wird zunächst mehrfach mit Blinddaten getestet, bevor er mit Realdaten ausgeführt wird.
- Einem Nutzer werden neue Projekte innerhalb der virtuellen Organisation zugewiesen.
- Basierend auf den Ergebnissen vergangener Grid-Jobs wird die Art der Berechnung verändert.
- Der Nutzer führt seine Jobs je nach Anforderungen auf verschiedenen Rechenressourcen aus.

Eine Gleichförmigkeit kann zwar unter Umständen im Einzelfall unterstellt werden, muss aber im Großen und Ganzen ausgeschlossen werden.

So ist auch die regelbasierte Einordnung des Nutzerverhaltens in legitimes und missbräuchliches Verhalten anhand der unterschiedlichen Nutzungsprofile schwierig. Was für einen Nutzer oder eine Nutzergruppe als absolut untypisches Verhalten gelten würde, etwa der regelmäßige Start sehr kurzer Jobs oder die Abgabe von zyklischen/rekursiven Jobs, ist für eine andere Nutzergruppe vollkommen normal. Es existieren keine Regeln für die Nutzung des Grid und die Formulierung von Grid-Jobs, auch „Best Practices“ sind nicht nennenswert vorhanden. Daher muss angenommen werden, dass für sich genommen praktisch keine Nutzungsart des Grid als legitim oder schädlich eingestuft werden kann.

Somit ist es für einen einzelnen Experten praktisch unmöglich, ein allgemein gültiges Regelwerk für das Nutzungsverhalten zu formulieren, das zur Missbrauchserkennung dienen kann. Ein solches müßte für jeden Nutzer individuell angelegt und stetig angepaßt werden. Außer dem Nutzer selber existiert jedoch kein Experte, der dessen Nutzungsverhalten kennt – und der Nutzer selber wird diesen Aufwand nicht betreiben wollen.

6.3.2 Fazit

Da es – wie im vorigen Abschnitt aufgezeigt – nicht praktikabel ist, ein umfassendes Regelwerk für das legitime Verhalten aller Grid-Nutzer aufzustellen, bleibt als einziger sinnvoller Ansatz die Nutzung eines wahrscheinlichkeitsbasierten oder heuristischen Ansatzes, der anhand von Trainingsdaten ein stochastisches Modell der Grid-Nutzung erstellt. Dieses Modell kann dann zur Missbrauchserkennung herangezogen werden. Es besteht zwar nach wie vor die Notwendigkeit, das Modell stetig an die sich ändernde Nutzungssituation anzupassen; diese Anpassung kann jedoch durch gezielte Kennzeichnung von Nutzungsdaten als Trainingsdaten geschehen und erfordert keine manuelle Erstellung neuer Regeln oder Regelwerke.

Aufgrund ihrer nachgewiesenen guten Eignung für den angestrebten Einsatzzweck (insbesondere ihrer Robustheit gegenüber veränderlichen Situationen, ihrer Skalierbarkeit und hohen Genauigkeit, wie in [LS94] ausgeführt) werden nun im Folgenden Bayessche Klassifikatoren für die weitere Analyse ausgewählt und als möglicher Algorithmus für die Missbrauchserkennung evaluiert.

6.4 Bayessche Netze und Klassifikatoren

Bayessche Netzwerke haben sich in den vergangenen Jahren in der praktischen Anwendung in der Informatik als robuste Möglichkeit zur wahrscheinlichkeitsbasierten Entscheidungsfindung durchgesetzt und wurden für so verschiedenartige Aufgaben wie die Modellierung von Fehlerfällen in Stromnetzen [He08] oder Spam-Erkennung in E-Mail-Systemen [SDHH98] verwendet.

Der Begriff „Bayessches Netzwerk“[2] geht auf den US-Informatiker Judea Pearl zurück, der in [Pea96] schrieb:

> „Bayesian networks are directed acyclic graphs (DAGs) in which the nodes represent variables of interest [..] and the links represent informational or causal dependencies among the variables.“

Ein Bayessches Netzwerk wird durch $B = \{G, P\}$ repräsentiert, wobei mit G den im vorigen Zitat erwähnten DAG (gerichtete azyklische Graph) und P eine Wahrscheinlichkeitstabelle bezeichnet. Jede gerichtete Kante zwischen zwei Knoten verdeutlicht einen probabilistischen Einfluss des Quellknotens (im folgenden „Vorgänger“) auf den Zielknoten. Unter der Annahme, dass jeder Knoten X_i nur von seinen Vorgängerknoten abhängig ist, kann eine Wahrscheinlichkeitsverteilung, die diese Voraussetzungen erfüllt, erstellt werden. Dazu wird für jeden Knoten X_i eine Tabelle mit bedingten Wahrscheinlichkeiten erstellt, die eine Wahrscheinlichkeitsverteilung über X_i für jede Variablenzuweisung seiner Vorgängerknoten beinhaltet.

Dieser Ansatz senkt die Anzahl notwendiger Parameter für eine multivariate Verteilung der Variablen – und bietet so eine effiziente Möglichkeit, die nachfolgenden Wahrscheinlichkeiten zu berechnen.

Um eine Anwendung von Bayesschen Netzwerken für die Missbrauchserkennung vorzubereiten, müssen demnach drei Schritte erfolgen:

1. Erstellung eines gerichteten azyklischen Graphen, der die Kommunikation im Grid modelliert.

2. Berechnung der Wahrscheinlichkeitstabellen für jeden Knoten.

3. Nutzung des resultierenden Netzwerks im Rahmen eines Bayesschen Klassifikators für die Missbrauchserkennung.

[2] benannt nach dem englischen Mathematiker Thomas Bayes (ca. 1702 - 1761)

Der zu erstellende gerichtete azyklische Graph orientiert sich an der Topologie der genutzten Grid-Knoten und Proxy-Credentials und kann anhand der eingehenden Audit Trails automatisch generiert werden. Schließlich sind für das Auditing zunächst nur jene Grid-Ressourcen relevant, zwischen denen eine Kommunikation mittels zu auditierender Proxy-Credentials stattgefunden hat; andere Knoten können zunächst vernachlässigt werden.

Die entsprechenden Wahrscheinlichkeitstabellen leiten sich aus der Häufigkeit der Kommunikation entlang der Kanten des DAG ab; sie sind daher bekannt oder können vor Auswertung durch die Maximum-Likelihood-Methode [Mur98] ermittelt werden.

Ein Bayesscher Klassifikator kombiniert sodann Bayessches Netzwerk sowie die ermittelten Wahrscheinlichkeiten und wird mithilfe eines Training-Datensatzes ermittelt.

6.4.1 Modellierung der Grid-Nutzung

In den meisten Anwendungsszenarien ist der grundsätzliche Ablauf eines Globus-Jobs recht gleichförmig:

1. Der Job wird auf einem UI-Server formuliert und gegebenenfalls getestet.
2. Der GRAM nimmt den Job zur Weiterverarbeitung an.
3. Eingabedaten werden auf den oder die genutzten Rechenknoten kopiert.
4. Der Job wird ausgeführt und berechnet mithilfe der Eingabedaten die Ergebnisdaten.
5. Die Ergebnisdaten werden an eine vom Nutzer spezifizierte Ressource zurückkopiert.

Für jeden Schritt in diesem skizzenhaften Arbeitsablauf wird ein Proxy-Credential zur Authentifizierung genutzt; die Ausführung des Jobs geht jeweils mit der Delegation einer oder mehrerer neuer Credentials einher; nur für den Datentransfer muss keine neue Delegation gebildet werden. Jede Nutzung zur Authentifizierung oder Delegation wird an den Auditing-Dienst gemelde. Dabei wird neben der Seriennummer des Zertifikats auch der empfangende Host (also der „Ort“, an dem der

Auditing-Datensatz generiert wird) gemeldet. Mittels dieser Informationen kann ein topologischer „Weg“ des Credentials durch das Grid abgebildet werden.

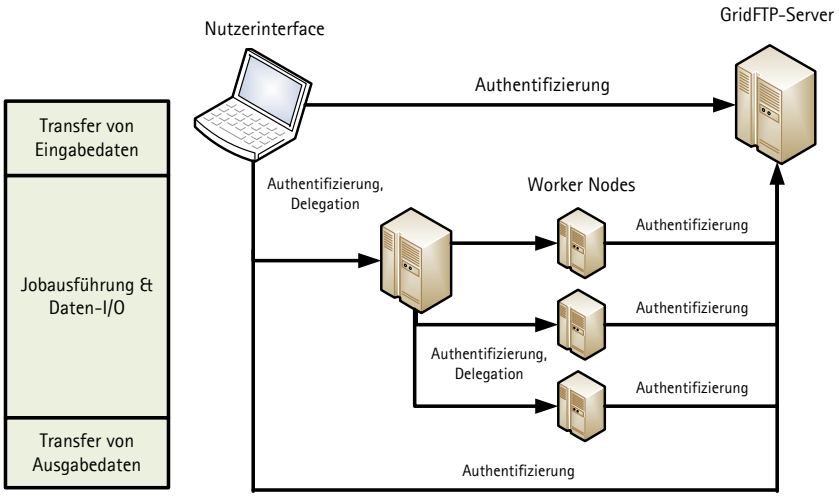
In einem gerichteten „Pfad-Graphen“, in dem jeder Knoten einem Grid-Host entspricht, wird die Topologie modelliert. Eine Kante zwischen zwei Knoten dieses Graphen existiert dann, wenn ein zu auditierendes Credential in einer Kommunikation zwischen diesen Ressourcen genutzt wurde. Abbildung 6.1 verdeutlicht diese Modellierung an einem Beispiel. Anhand der durch das Auditing gesammelten Nutzungsdaten kann für jedes Paar von Knoten P,Q im Grid die Wahrscheinlichkeit ermittelt werden, dass eine Kommunikation $P \rightarrow Q$ oder $Q \rightarrow P$ stattgefunden hat.

Ein zweiter Graph wird zur Modellierung der Ableitungspfade von Grid-Credentials herangezogen; dieser Graph wird als „Credential-Graph“ bezeichnet. Er hat eine Baumstruktur, dessen Wurzel das erste Proxy-Credential bildet, also dasjenige Zertifikat, das mit dem privaten Schlüssel des End-Entity-Zertifikats des Nutzers erstellt wurde. Für jedes direkt oder indirekt von diesem „Proxy erster Ordnung“ abgeleitete Credential wird ein neuer Knoten in den Baum hinzugefügt; eine Kante verbindet dieses mit dem erzeugenden oder „Eltern-“Credential. Dieser in Abschnitt 2.2.4 geprägten Terminologie folgend, werden zwei vom selben Elternteil erzeugte Credentials als „Geschwister“ bezeichnet. Wie auch für den zuvor erzeugten Pfad-Graphen kann mithilfe der erhobenen Auditingdaten eine Wahrscheinlichkeit für die Anzahl an Vor-, Nachfahren oder Geschwistern für Proxy-Credentials ermittelt werden.

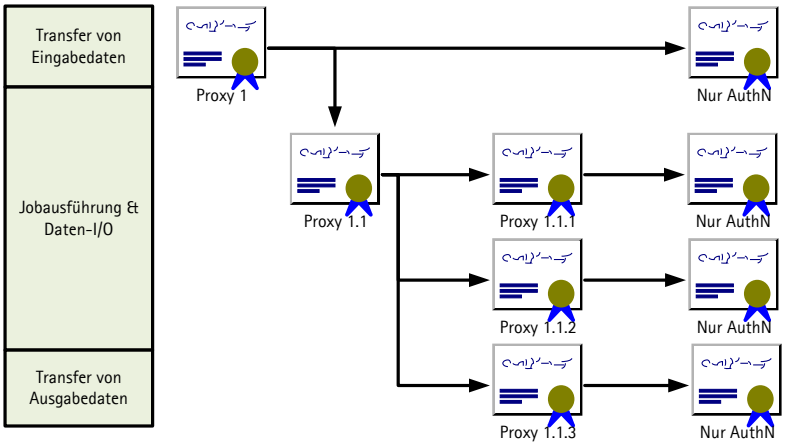
Beide Graphen kombiniert können als Indikator für Missbrauch dienen und erfüllen die formalen Kriterien für ein bayessches Netzwerk. Da diesem ein Graph zugrunde liegt, der gerichtet und azyklisch ist, muss jedoch zunächst noch geprüft werden, ob diese beiden Bedingungen erfüllt sind.

Eine Richtung erhält jede Kante des Pfad-Graphen dadurch, dass (wie oben erläutert) sie eine Kommunikation zwischen zwei Knoten modelliert und damit ein Initiator der Kommunikation – der Ursprung der Kante – und ein Rezipient, der das Ziel der gerichteten Kante ist, existiert.

Ebenso sind die im Credential-Graphen abgebildeten Vorgänge der Authentifizierung und Delegation jeweils gerichtet. Die Authentifizierung wird, obgleich gegenseitig, von einem der beiden Kommunikationspartner initiiert; ebenso wird bei der Delegation zwischen Delegationsgeber und -empfänger unterschieden. Während



(a) Pfad-Graph



(b) Credential-Graph

Abbildung 6.1: Beziehungen zwischen Ressourcen und Credentials während eines Grid-Jobs

der Delegationsgeber über das Credential nebst dem Private Key verfügt, von dem abgeleitet wird, erhält der Delegationsempfänger ein signiertes Proxy-Credential.

Die Azyklizität des Pfadgraphen nachzuweisen, gelingt unter Berücksichtigung der Kombination beider Graphen. Betrachtet man die Topologie des Grid ohne jede Berücksichtigung des Ableitungsmechanismus bei Proxies, so stellt man fest, dass eine Zyklizität in der Kommunikation zwischen verschiedenen Knoten über die im Graphen vorgesehenen Kanten berücksichtigt werden muss, wie etwa bei der Kommunikation zwischen Job-Scheduler (GRAM) und Rechenknoten. Obgleich sich so zwischen zwei oder mehr Knoten zyklische Abläufe bilden können (die in einem Bayesschen Netzwerk nicht modelliert werden können), stellt sich die Situation bei Berücksichtigung der in dieser gerichteten Kommunikation verwendeten Proxy-Credentials anders dar. Für jede Kommunikation entlang der zyklischen Kanten wird ein unterschiedliches Credential abgeleitet und eingesetzt, so dass die Azyklizität des im bayesschen Netzwerk modellierten Graphen garantiert werden kann.

6.5 Modellierung des Klassifikators

Mittels im „Pfad-Graph“ und im „Credential-Graph“ modellierten Wissens kann nun ein bayesscher Klassifikator (Classifier) erzeugt werden. Beide Graphen werden zunächst miteinander korreliert (was anhand der topologischen Information, auf welchem Knoten im Grid ein Credential zur Authentifizierung oder Delegation genutzt wurde, möglich ist).

Der resultierende erweiterte Credential-Graph enthält somit neben der eindeutigen ID jedes Proxy-Credentials (*cID*) noch Informationen über die Anzahl der Vorfahren (*#vor*), die Anzahl der parallel erzeugten „Geschwistercredentials“ (*#geschw*), Informationen über den im Grid genommenen Pfad (*gridPfad*) sowie die eindeutige Identifikation des Credential-Besitzers (*userID*). Jeder Knoten des erweiterten Graphen enthält also das folgende Tupel: (*cID, userID, #vor, #geschw, gridPfad*). Ein solches Tupel kann für jeden Audit Trail, der in der Simulation eines auditingfähigen Grids (siehe Kapitel 6) oder mittels der prototypischen Implementierung (siehe Abschnitt 5.8) generiert wurde, gebildet werden. Es formt die Grundlage, auf der nun mittels der Software WEKA ein bayesscher Klassifikator gebildet werden kann.

6.5.1 Klassifikation mittels WEKA

Das Programm WEKA (Waikato Environment for Knowledge Analysis) [HFH+09] wurde im Rahmen eines von der neuseeländischen Regierung geförderten Projekts im Jahr 1992 entwickelt. Es gilt seitdem als eines der wichtigsten Werkzeuge zur Anwendung von Algorithmen des maschinellen Lernens. In einer interaktiven „Workbench"-Umgebung können Algorithmen und Datensätze evaluiert werden; die modulare Architektur erleichtert die Implementation neuer Funktionen.

Mittels vorher generierter Datensätze in einem WEKA-eigenen Format, welche die im letzten Absatz erwähnten fünf Merkmale enthalten, wurde zunächst die Struktur eines bayesschen Netzwerks ermittelt, das zur Modellierung der in den Testdaten enthaltenen Grid-Nutzung geeignet ist. Danach wurde eine Maximum-Likelihood-Methode (wie in [Mur98] beschrieben) angewendet, um die notwendigen Wahrscheinlichkeitswerte für jeden Knoten des Graphen zu ermitteln. Somit lag ein bayessches Netzwerk vor, das zur Klassifikation der Grid-Nutzung herangezogen werden konnte.

Detaillierte Informationen über den Einsatz der WEKA-Software bei der Simulation eines Grid und die Ergebnisse der Klassifikation findet der Leser in Kapitel 6 respektive Abschnitt 7.4.1.

6.6 Notwendigkeit einer Simulation

Für die im Rahmen dieser Dissertation erarbeitete Lösung zum Auditing von Proxy-Zertifikaten wurde eine prototypische Implementation vorgestellt (siehe 5). Diese ist geeignet, die Nutzung von Grid-Diensten in einem Globus Container (also u.a. des WS-GRAM) zu auditieren und wurde in einer Testumgebung erfolgreich erprobt.

Die Idee der Missbrauchserkennung mittels bayesscher Klassifikatoren bedingt jedoch, dass Trainingsdaten in ausreichendem Maße zur Verfügung stehen, und zwar sowohl für legitime als auch für missbräuchliche Grid-Nutzung. Diese Trainingsdaten können in einer Testumgebung nicht sinnvoll generiert werden, da es sowohl an Variation (also verschiedenen Nutzungsmustern) als auch an Nutzungsvolumen (also verschiedenen Komponenten und Nutzern) mangelt. Nur eine reale Grid-Umgebung mit ausreichender Nutzerzahl kann diese Anforderungen gewährleisten und auch hier ist eine längere Phase der Messung und Verarbeitung von Trainingsdaten notwendig.

Eine neuartige Anwendung wie das Auditing, die zudem in die dem Grid zugrundeliegenden Sicherheitsmechanismen eingreift, kann jedoch in einer produktiv

genutzten Grid-Infrastruktur zum Einsatz kommen, ohne zuvor gründlich getestet worden zu sein. Somit ergibt sich ein Dilemma, da für eine Erprobung des Projekts wichtige Nutzdaten fehlen, diese jedoch nicht ohne Weiteres gewonnen werden können. Um die Ansätze automatisierte Erkennung missbräuchlicher Grid-Nutzung dennoch validieren zu können, wurde zusätzlich zum Prototyp eine Simulation durchgeführt, die eine variantenreiche Grid-Nutzung – sowohl legitimer als auch missbräuchlicher Natur – simuliert und somit umfangreiche Trainingsdaten zur Erstellung des bayesschen Netzes und Klassifikators generiert.

6.7 Grid-Simulation mit WEKA

Wie bereits im vorigen Abschnitt erwähnt, liegt der Fokus der Simulation in einer möglichst hohen Varianz der Grid-Nutzung. Diese Varianz ist zum einen eine Abbildung der Grid-Nutzung verschiedener Nutzergruppen (also verschiedener wissenschaftlicher Disziplinen), zum Anderen auch geeignet, um die sich ändernden Nutzungsmuster einzelner Grid-Nutzer zu simulieren.

Die exemplarische Simulation einer auditingfähigen Grid-Infrastruktur wurde in fünf Experimenten durchgeführt, die jeweils verschiedene Nutzungs- und Nutzerprofile enthielten. In jedem Experiment wurden sechs verschiedene Gruppen mit jeweils derselben Nutzeranzahl simuliert. Diese Nutzergruppen bilden jeweils eine virtuelle Organisation oder wissenschaftliche Community ab; Mitglieder einer Gruppe zeigen jeweils ähnliches Nutzungsverhalten. Die Möglichkeit einer Überlappung der Gruppenmitglieder – eine VO-Mehrfachmitgliedschaft ist in realen Grid-Infrastrukturen durchaus üblich, wenn ein Nutzer zum Beispiel in mehreren Projekten mitarbeitet oder neben wissenschaftlichen noch Lehraufgaben erfüllt – wird in der Simulation ebenfalls abgebildet.

Basierend auf dem jeweiligen Profil einer VO wurden die Nutzerprofile zufallsgesteuert erstellt, und danach entsprechende Auditing-Datensätze generiert. Die Wahrscheinlichkeit, dass ein bestimmtes Credential durch Authentifizierung oder Delegation an einer Kommunikation zwischen zwei Knoten beteiligt ist, wird ebenso im Profil des Besitzers festgelegt wie die Wahrscheinlichkeit einer Ableitung mehrerer „Kinder“ vom selben Ursprungszertifikat.

Die maximale Anzahl von Delegationen (also die Länge der Zertifikatskette) und die maximale Anzahl von „Geschwistern“ wird im VO-Profil festgelegt und variiert

zwischen verschiedenen Nutzergruppen, nicht aber zwischen Nutzern einer Nutzergruppe. Diese Entscheidung wurde aufgrund der Annahme getroffen, dass die in diesen Parametern abgebildete Nutzungsvariation sich nur zwischen Nutzern verschiedener VOs unterscheidet.

Für jeden simulierten Nutzer wird nun basierend auf den im VO- und Nutzerprofil festgelegten Informationen eine zufällige Zahl von Grid-Jobs submittiert; die Maximalzahl der Jobs pro Nutzer wird als Parameter des Experiments definiert.

6.7.1 Simulation missbräuchlicher Grid-Nutzung

Eine Schwierigkeit der Erprobung des Auditing in einer produktiven Grid-Umgebung ist die Erhebung von Trainingsdaten aus Mißbrauchsfällen. Ein Set von Trainingsdaten muss auch Beispiele illegitimer Nutzung enthalten, die nicht nach Bedarf in ausreichendem Maße generiert werden können, ohne missbräuchliche Handlungen selbst zu begehen.

In der durchgeführten Simulation wurden ausreichend Testdaten missbräuchlicher Nutzung generiert, um dieses Problem zu umgehen. Diese wurden anhand realer Bedrohungsszenarien (siehe auch 3.4 für detalliertere Informationen zu den möglichen Angriffsszenarien) modelliert und decken zwei der häufigsten Missbrauchsfälle ab:

1. Die nach einem Credentialdiebstahl mögliche missbräuchliche Ressourcennutzung wird modelliert, indem von einem Credential eine extrem große Anzahl paralleler Ableitungen generiert wird, die dann zur Abgabe von Grid-Jobs genutzt werden. Zusätzlich wird die Wahrscheinlichkeit für zyklische Verwendung von Delegationen[3] bei diesem Szenario erhöht.

2. Das schnelle „Scannen“ von Datenressourcen auf für einen Angreifer interessante Datensätze wird im zweiten Missbrauchsszenario modelliert, indem mit demselben Proxy-Credential in schneller Folge eine Kommunikation zu vielen verschiedenen Grid-Knoten, insbesondere Datenknoten, stattfindet.

Mit diesen vereinfachten Modellen der Grid-Nutzung und der damit einhergehenden auditierbaren Authentifizierung und Autorisierung konnte eine bedarfsgerechte Abbildung der Realität vorgenommen werden, die als Testumgebung für die im Rahmen dieser Dissertation entwickelten bayesschen Klassifikatoren dient.

[3]Zyklische Verwendung liegt vor, wenn ein Proxy-Credential auf dem Host zur Abgabe von Jobs genutzt wird, auf dem sein Elternzertifikat erzeugt wurde

6.8 Evaluation

Die Erkennung missbräuchlicher Credentialnutzung stellt einen wichtigen Teil des Gesamtkonzeptes dar. Ein Verzicht auf diese Komponente hätte den Nutzwert des Projektes geschmälert und war daher zu vermeiden. Dennoch stellte sich während der Recherchen für mögliche Erkennungsmechanismen heraus, dass ein Ansatz aus dem „Machine Learning“ ohne eine ausreichend große Datenbasis nicht möglich sein würde. In Abschnitt 6.6 sind die Schwierigkeiten bei der Erhebung einer solchen Datenbasis beschrieben – die Entscheidung, die Missbrauchserkennung zunächst in einer Simulation zu approximieren, gründet primär auf der Unmöglichkeit, echte Nutzdaten zu erheben.

Die Simulation starker Grid-Nutzung anhand zuvor definierter Modelle und Nutzergruppen nähert sich der realen Nutzung in einem nationalen Grid mit vielen Nutzern gut an und kann durch die große Flexibilität der Simulation auch extreme Randfälle gut abdecken. Sie liefert somit wertvolle Indikationen, ob eine automatisierte Missbrauchserkennung von Grid-Proxy-Credentials möglich und mit welchen Fehlerquoten zu rechnen ist.

In Kapitel 3 wurden spezifische Angriffsmuster skizziert, wie sie durch inkorrekte „Garbage Collection“ und Ausnutzen von Systemlücken auf einer Ressource entstehen können. Diese Szenarien dienten als Grundlage für die vorgenommene Simulation und können durch die Komponente zur Missbrauchserkennung detektiert werden. Insbesondere ist zu erwähnen, dass die in Abs. 3.5.3 und 3.5.1 beschriebenen Folgen eines erfolgreichen Angriffs deutliche Indikatoren für das Auditing liefern.

Werden Ressourcen missbräuchlich genutzt, so erfolgt dies typischerweise mittels massenhafter Job-Submission durch den Angreifer; dieser ist sich schließlich des begrenzten Zeitfensters bewußt, das ihm zur Verfügung steht. Die massenhafte parallele Abgabe von Grid-Jobs ist eines der beiden in 6.7.1 beschriebenen simulierten Szenarien; es zeichnet sich durch eine hohe Anzahl paralleler Ableitungen (je eine pro abgegebenem Grid-Job) aus, die als zuverlässiges Indiz für missbräuchliche Grid-Nutzung gelten.

Beabsichtigt ein Angreifer, die Daten des Nutzers zu manipulieren, so wird er zunächst kurz auf alle zur Verfügung stehenden Datenressourcen (etwa auf Basis eines öffentlichen Verzeichnisses) zugreifen, eine Liste der vorhandenen Dateien abrufen und dann gezielt jene manipulieren, die er als attraktive Ziele identifiziert

hat. Dieses Zugriffsmuster benötigt in der Regel keine zusätzlichen Delegationen[4] (ansonsten wäre es auf dieselbe Art und Weise erkennbar wie die missbräuchliche Job-Submission im letzten Absatz), unterscheidet sich aber von regulären Zugriffen durch eine rasch aufeinanderfolgende kurze Kommunikation mit vielen verschiedenen Datenressourcen. Auch dieses Muster wird verlässlich erkannt.

Die Missbrauchserkennung selber, wie sie prototypisch anhand der simulierten Daten durchgeführt wurde, stützt sich ausschließlich auf die während des Auditing erhobenen Daten und kann – sofern ausreichend simulierte Datensätze vorliegen – eine sehr zuverlässige Einschätzung liefern. Der entscheidende Faktor für eine produktive Nutzung ist jedoch, ob und wie reale Missbrauchsdaten erhoben werden können; es ergibt sich hier eine Art „Cold Start Problem“ [SPUP02]. Es könnte hier notwendig werden, anhand einer definierten Liste möglicher Missbrauchsfälle in einer kontrollierten Umgebung Missbrauchsdaten zu generieren (etwa durch die Administratoren von Grid-Ressourcen), um vom Erkennungsalgorithmus verwertbare Daten zu erhalten.

6.8.1 Ergebnisse der Simulation

Jedes der fünf durchgeführten Experimente resultierte in einer – aufgrund der zufallsgesteuerten Abfolge der Simulation stark variierenden – Anzahl von Auditing-Datensätzen. Im Durchschnitt wurden pro Experiment etwas über 1,4 Millionen Auditing-Tracks erhoben; bei einem Minimum von 379 und einem Maximum von 6.7 Millionen Datensätzen im umfangreichsten Experiment.

Zusätzlich variiert auch der Prozentsatz missbräuchlicher Nutzung. Die Missbrauchsquote ist in solchen Experimenten sehr niedrig, die einen langen Nutzungszeitraum simulieren, an dessen Ende ein Angriff erfolgt. Andere Simulationen enthalten hingegen mehrere aufgezeichnete Angriffe. Im Schnitt waren 4,5% der Datensätze in den Experimenten missbräuchlicher Nutzung zuzuordnen.

Die in der Simulation gewonnenen Daten wurden nach einer Trainingsphase zur Erkennung in den bayesschen Klassifikator eingegeben. Die resultierenden Erkennungswahrscheinlichkeiten werden insbesondere an der Genauigkeit und der Trefferquote gemessen. Beide Werte wurden für die zwei möglichen Erkennungsklassen „legitime Nutzung“ und „Missbrauch“ separat berechnet. Die Trefferquote (Preci-

[4] GridFTP und OGSA-DAI benötigen keine Delegationen für den Datenabruf von Nicht-GSI-Backends

sion), bezeichnet den Quotienten aus allen Datensätzen, die tatsächlich einer Erkennungsklasse angehören und jenen Datensätzen, die vom Klassifikator als jener Klasse zugehörig eingeordnet wurden. Sie ist wie folgt definiert:

$$\text{precision} = \frac{tp}{(tp+fp)}$$

Hier bezeichnet tp die Anzahl korrekt erkannter Datensätze („true positives“) und fp die „false positives“, also die fälschlich erkannten Datensätze in der betreffenden Erkennungsklasse.

Die Genauigkeit (Recall) ist der Quotient aus den Datensätzen, die (korrekt oder inkorrekt) als missbräuchlich oder legitim erkannt wurden sowie aller Datensätze, die dieser Erkennungsklasse wirklich angehören. Sie ist definiert als:

$$\text{recall} = \frac{tp}{(tp+fn)}$$

Die Bedeutung der Variablen ergibt sich analog zur Definition der Trefferquote; fn bezeichnet die „false negatives“, also die fälschlich nicht erkannten Datensätze der Erkennungsklasse.

Die durchschnittliche Genauigkeit und Trefferquote sind recht hoch. So beträgt die Trefferquote 99,5%, die Genauigkeit beträgt ebenfalls 99,5% – beide Zahlen wurden auf eine Nachkommastelle gerundet.

Zwei weitere wichtige Maßzahlen für die Erkennungsgenauigkeit sind die Falscherkennungsquoten für missbräuchliche und legitime Nutzung. Der Prozentsatz der „False Positives“, also fälschlich als missbräuchlich eingestufte legitime Grid-Nutzung, beträgt 0,4%. Eine irrtümlich als legitim erkannte, jedoch in Wahrheit missbräuchliche Nutzung lag in 0,7% der Fälle vor.

6.9 Fazit

Die automatische Erkennung von Missbrauch bei Grid-Proxy-Credentials ist mittels bayesscher Klassifikatoren mit einer hohen Genauigkeit möglich. Die in einer umfangreichen Simulation generierten Daten wurden mit besonderem Fokus auf eine möglichst hohe Varianz in der Grid-Nutzung erhoben, um alle Eventualitäten in einer großen Grid-Umgebung berücksichtigen zu können. Die erzielte Erkennungsgenauigkeit von 99,5% unterstreicht die Gültigkeit des Ansatzes, wahrscheinlichkeitsbasierte „Belief Networks“ für die Missbrauchserkennung zu verwenden.

Kapitel 7

Evaluation

Im Rahmen einer Evaluation des Gesamtkonzepts wird im folgenden Kapitel nachgewiesen, daß durch das Auditing von Grid-Proxy-Credentials für den Nutzer ein echter Mehrwert generiert wird. Dazu werden alle bisher vorgestellten Teilkomponenten einer kritischen Würdigung unterzogen.

7.1 Einleitung

In verteilten Infrastrukturen wie dem in dieser Dissertation exemplarisch behandelten D-Grid werden Proxy-Zertifikate im Rahmen einer X.509-PKI verwendet, um die eine zuverlässige Authentifizierung von Nutzern und Ressourcen zu garantieren und um Zugriffsrechte delegieren zu können, ohne „wertvolle“ Zugriffstokens aus der Hand geben zu müssen. In Kapitel 3 wurde aufgezeigt, dass diese Praxis Vertrauen nicht nur in die Zertifikatsstelle (CA), sondern auch in die Administratoren und Teilnehmer einer Grid-Infrastruktur voraussetzt und dass erfolgreiche Angriffe den Missbrauch der ungesicherten Proxy-Credentials nach sich ziehen können. Gleichzeitig fehlte Nutzern eine Möglichkeit, die Nutzung dieser Credentials zu überwachen und somit Missbrauch rechtzeitig zu erkennen.

Vorhandene Komponenten im Globus Toolkit sind nicht in der Lage, die Nutzung von Credentials zentralisiert und zuverlässig aufzuzeichnen und kommen somit nicht als Lösungsmöglichkeit für dieses Problem in Betracht. Es wurde im Rahmen dieser Dissertation also eine neue Komponente für das Globus Toolkit entworfen, die die Nutzungsverfolgung – das Auditing – von Proxy-Credentials ermöglicht.

7.2 Evaluation des Gesamtkonzepts

Die in Kapitel 3 durchgeführte Analyse der Sicherheitslage in einer großen Grid-Infrastruktur (dem D-Grid) ergab, dass die Verwendung von Proxy-Credentials zur Authentifizierung und Delegation von Benutzerrechten inhärente Sicherheitsprobleme mit sich bringt, deren Lösung jedoch ohne eine Abkehr vom Delegationskonzept des Grid nicht machbar erscheint. Zudem ergab die Analyse, dass für Nutzer des Grids nicht ersichtlich ist, welche Ressourcen mittels delegierter Credentials in ihrem Namen handeln.

Das in Kapitel 4 vorgestellte Konzept für eine Grid-Auditing-Infrastruktur legt nun den Grundstein für eine zuverlässige und sichere Möglichkeit, die Nutzung von „Grid-Assets“ zu protokollieren und möglichen Missbrauch festzustellen. Insbesondere die Verwendung von Grid-Proxy-Credentials ist mit der vorgestellten Architektur deutlich nachvollziehbarer und somit transparenter geworden.

Der Entwurf des Auditing-Systems erfolgte von vorneherein mit Hinblick auf bestmögliche Flexibilität, aber auch unter Berücksichtigung wichtiger weiterer Kriteri-

en wie der Interoperabilität mit dem Grid und Sicherheit der Gesamtlösung. Es entstand ein standardkonformes System, das auf bereits bestehenden Infrastrukturkomponenten aufsetzt und diese insofern erweitert, dass Informationen (sog. „Audit Tracks“) über die Nutzung von Proxy-Zertifikaten von der modifizierten Komponente an einen zentralisierten Webservice versandt werden. Dieser Versand wahrt die Sicherheitsanforderungen an Integrität, Authentizität und Vertraulichkeit der Audit Tracks und geschieht in einem grid-üblichen XML-basierten Datenformat.

Ein Webservice empfängt diese Datensätze und speichert sie zunächst in einer nachgelagerten relationalen Datenbank. Diese unterstützt durch ihr Tabellenkonzept eine breite Menge an Datentypen und -modellen.

Durch den modularen Aufbau des Systems ist dieses zur Auditierung der in dieser Dissertation hauptsächlich als Einsatzzweck genannten Proxy-Credentials geeignet – alle Systemkomponenten können jedoch auch zur Erfassung anderer Informationen wie etwa den Zugriffen auf Daten verwendet werden. Die starke Verankerung von Industriestandards in Konzept und Referenzimplementation (siehe dazu 7.4) erlaubt auch einen Einsatz außerhalb von Grid-Infrastrukturen etwa in Clouds oder im Rahmen anderer Authentifizierungsmechanismen mit einer Möglichkeit der Rechtedelegation.

In dieser Dissertation wurde jedoch als wichtiges Anwendungsbeispiel die Auditierung von Grid-Proxy-Credentials genannt; die Evaluation des Konzeptes befasst sich im Folgenden demnach auch mit dieser Einsatzmöglichkeit.

Grid-Infrastrukturen setzen auf dezentrale Mechanismen zur Authentifizierung und Delegation von Rechten; eine X.509-PKI wird zu diesem Zweck eingesetzt. Im Entwurf dieser Dissertation war es daher notwendig, eine Auditierungsmöglichkeit zu finden, die auch in einer dezentralen Umgebung wie dem Grid einsetzbar ist. Außerdem ergab die Analyse bestehender Komponenten, dass keine davon erweitert oder genutzt werden konnte.

Die Grid Security Infrastructure (GSI) liegt als programmatische Basis jeder Grid-Ressource zugrunde und verfügt über die für eine Auditierung notwendigen Informationen. Es lag daher nahe, sie als Ansatzpunkt für die Konzeption des in dieser Arbeit vorgestellten Systems zu verwenden. Durch diese Modifikation konnte die Möglichkeit zur Auditierung in jede Infrastrukturkomponente integriert werden, ohne jeweils eine eigene Schnittstelle zu implementieren. So wurden nicht nur mögliche Fehlerquellen in implementatorischer Hinsicht vermieden, sondern auch die Einheit-

lichkeit der Auditingdaten sichergestellt. Desweiteren erfüllt dieser Ansatz auch das Kriterium der Vollständigkeit, wie es in 4.2 gefordert wurde: Sämtliche Komponenten des Globus Toolkit greifen auf die Java- oder C-Bibliotheken der GSI zurück und unterstützen somit ein Auditing, sobald es in jenen Bibliotheken enthalten ist.

Da ein Proxy-Credential stets digital signiert ist (entweder mithilfe des End-Entity-Credentials des Nutzers oder eines anderen Proxy-Credentials), stellt die X.509-Erweiterung auch eine valide und nicht abstreitbare Willensäußerung des Zertifikatsinhabers dar. Dieser kann sich somit darauf verlassen, dass – sofern eine Grid-Ressource auditingfähig ist und nicht Opfer eines gegen die Auditing-Infrastruktur gerichteten Angriffs[1] wurde – seinem Wunsch nach Auditierung des Credentials auch dann entsprochen wird, wenn das Credential in falsche Hände geraten ist. Die Zertifikatserweiterung kann ohne Zerstören der digitalen Signatur – und somit Invalidierung des gesamten Proxy-Credentials – nicht entfernt oder überschrieben werden.

Die Kompatibilität und Interoperabilität mit bestehenden Komponenten gehört zu den wichtigsten Anforderungen, die in Abschnitt 4.2 identifiziert wurden. Diese Interoperabilität gilt insbesondere auch für solche Komponenten, die nicht auditingfähig sind. Deren Betrieb darf durch das Auditing nicht negativ beeinflusst werden, weswegen alle Modifikationen so gestaltet wurden, dass Auditing grundsätzlich optional ist. Eine Zertifikatserweiterung, die von nicht auditingfähigen Ressourcen ignoriert werden kann, dient als Auslöser für die Nachverfolgung auf denjenigen Komponenten, deren GSI-Bibliotheken ein Auditing durchführen können. Durch diese Optionalität ist die Kompatibilität mit nicht auditingfähigen Grid-Komponenten sichergestellt; gleichzeitig hat der Nutzer (der die Zertifikatserweiterung durch eine entsprechende Option bei der Zertifikatserstellung aktiviert) stets die Kontrolle über das Auditing seiner Credentials.

Maximale Flexibilität bei der Speicherung der angefallenen Datensätze wird durch eine relationale Datenbank (siehe Abschnitt 4.4.4) erreicht. Das zugrundeliegende Datenbankdesign kann modular an sich ändernde Anforderungen angepaßt werden – etwa, wenn zusätzliche Informationen auditiert werden sollen – und ermöglicht so zukünftige Erweiterungen des Konzepts.

Die wichtigste in dieser kritischen Betrachtung zu beantwortende Frage ist jedoch: Ist das vorgestellte Konzept geeignet, mehr Transparenz und Sicherheit bei der Verwendung von Grid-Proxy-Credentials zu schaffen?

[1] siehe hierzu Abschnitt 7.3

Um diese Frage zu beantworten, sei zunächst ein kurzer Rückblick auf die Ausgangssituation gestattet. In einer globus-basierten Grid-Umgebung erzeugt der Nutzer ein Credential und gibt dieses ins Grid ab – entweder durch Nutzung bei der Jobabgabe oder mittelbar über ein System wie etwa MyProxy. Sobald der Nutzer die Kontrolle über seine Delegation abgibt, werden ihm keinerlei Informationen dazu mehr übermittelt. Er muss also darauf vertrauen, dass alle Komponenten des von ihm genutzten Grids die ihnen anvertrauten Credentials korrekt nutzen, ein Nachweis für dieses Vertrauen kann aber nicht erbracht werden. Lediglich die in Abschnitt 3.2 erwähnten Mechanismen zur Nachverfolgung von Jobs stehen ihm zur Verfügung; diese klammern aber prinzipbedingt die genutzten Credentials vollständig aus.

Im Rahmen dieser Arbeit wurde nun die GSI insofern erweitert, als dass jede Nutzung eines Proxy-Credentials auf einer Grid-Ressource einen Eintrag bei einem Auditing-Dienst hinterlässt, der nachträglich nicht mehr geändert werden kann (siehe Abs. 4.4.4). Dieser Auditing-Eintrag stellt eine neue Information zur Verfügung, er liefert also mithin bereits in dieser unbearbeiteten Form einen Mehrwert. Durch die in einem Auditing-Record erhobenen Daten ist es zudem möglich, mehrere logisch zusammenhängende Records – die sog. „Audit Trails“ – zu aggregieren und so weitere Informationen zu erhalten. Pragmatischerweise werden Audit Trails gebildet, indem von einem Ursprungs-Credential aus alle Authentifizierungs- und Ableitungsvorgänge, die auf diesem Credential basieren, aggregiert werden. Für den Nutzer ergibt sich somit eine hohe Transparenz, da er den „Weg“ jedes seiner Credentials durch das Grid praktisch lückenlos nachvollziehen und diese Verfolgung auch über mehrere untergeordnete Ableitungsvorgänge hinweg aufrechterhalten kann. Der Nutzer ist somit voll darüber informiert, auf welchen Ressourcen seine Credentials vorgehalten und genutzt werden.

Mittels dieser für den Nutzer zugänglichen Aggregation von Auditing-Daten der Proxy-Credentials wurde die Transparenz der Zertifikatsnutzung im Grid bereits wesentlich erhöht. Die zuvor gestellte Frage kann somit in Bezug auf die geforderte Transparenz positiv beantwortet werden.

Durch die zusätzliche Funktionalität einer Komponente zur Missbrauchserkennung wird zudem die effektive Sicherheit erhöht, da der Nutzer – anders als in konventionellen Grid-Umgebungen – nun über eine Möglichkeit verfügt, Missbrauch zuverlässig (vgl. die separate Evaluation in Abs. 7.4.1) zu erkennen und entsprechende Maßnahmen zu ergreifen. Eine solche Möglichkeit zur Missbrauchserkennung fehlt

in aktuellen Grid-Infrastrukturen vollständig und ist als vorbereitende Maßnahme zur Missbrauchsvermeidung etwa durch den Rückruf von Credentials unerlässlich.

Wie bei jedem anderen Sicherheitssystem muss jedoch auch in dem vorliegenden Konzept damit gerechnet werden, dass Angriffe mit dem Ziel unternommen werden, das System außer Funktion zu setzen oder anderweitig zu unterwandern. Einige Angriffsszenarien gegen das Auditing-System, mögliche Gegenmaßnahmen und dafür notwendige Erweiterungen werden im folgenden Abschnitt evaluiert.

Fazit

Insgesamt ist die dieser Arbeit zugrundeliegende Problemstellung, nämlich die fehlende Transparenz beim Umgang mit Proxy-Credentials im Grid, durch das vorgestellte Konzept umfassend gelöst worden. Nutzer, die vom Proxy-Auditing Gebrauch machen, erhalten einen umfassenden Bericht über die Nutzung der von ihnen delegierten Credentials im Grid, auch wenn von diesen weitere Ableitungen erstellt werden. Die separat evaluierte Komponente zur automatischen Missbrauchserkennung stellt eine sinnvolle Ergänzung dar und bietet Nutzern die Möglichkeit, auch ohne spezielles Fachwissen die wichtigsten Informationen über möglichen Missbrauch ihrer Credentials zu erhalten.

7.3 Widerstandsfähigkeit gegen Ausfälle

Auch ohne einen dezidierten Angriff können wie in jedem anderen IT-System Ausfälle auftreten, die die Funktionsfähigkeit der Auditing-Infrastruktur beeinflussen können. Insbesondere sind hier Ausfälle eines Auditing-Dienstes, der Datenbankinfrastruktur oder der Kommunikationswege zwischen den Komponenten zu nennen. Das in der vorliegenden Arbeit konzipierte Gesamtsystem ist jedoch so ausgelegt, dass es ausfalltolerant agieren kann. Fällt ein Auditing-Dienst oder ein Kommunikationsweg zwischen einer Grid-Komponente und dem Auditing-Dienst aus, so kann über den oben erläuterten Mechanismus mehrerer Dienste-URLs auf einfache Art und Weise eine Redundanz geschaffen werden; nach einem konfigurierbaren Timeout wird ein Auditing-Record an einen sekundären Dienst gesandt.

Ist der ursprüngliche Auditing-Dienst wieder verfügbar, so kann er mittels eines GSI-gesicherten Kommunikationskanals die zwischenzeitlich aufgelaufenen Auditing-Tracks vom sekundären Dienst abrufen; ein Protokoll für Authentifizierung und Au-

torisierung dieses Abrufs ist in einer zukünftigen Arbeit zu entwerfen.

Selbst ein Totalausfall der Auditing-Infrastruktur würde die umgebende Grid-Infrastruktur nicht wesentlich negativ beeinflussen, da das Auditing als optionales Zusatzfeature implementiert ist. Kann kein Auditing durchgeführt werden, so wird die nicht auditierbare Kommunikation zugelassen; ein Kommunikationsabbruch findet nicht statt („fail open“).

Die verwendeten Subsysteme des Auditing-Systems sind leicht mit Mitteln der jeweiligen Software gegen Ausfälle schützbar. So können der Globus Container, aber auch das Datenbanksystem, auf mehrere Hostmaschinen repliziert werden; eine Synchronisationsmöglichkeit steht von Haus aus zur Verfügung. Mit den Mitteln des konventionellen Unix-Systemmanagements kann so leicht ein hochverfügbares Auditing-System implementiert werden, dessen Ausfallwahrscheinlichkeit sehr gering ist.

7.4 Evaluation der Implementation

Im Rahmen einer Masterarbeit wurde – wie in Kapitel 5 ausführlich erläutert – eine Implementation der Auditing-Komponenten für den webservice-basierten Teil des Globus Toolkit 4 erstellt. Diese Implementation deckt die wichtigsten Funktionen des Auditing-Konzepts ab und dient somit als „Proof of Concept“.

In Kapitel 4 wurden verschiedene Kriterien aufgestellt, die durch das Auditing-Konzept erfüllt werden mussten. Diese Kriterien sind entsprechend auch für die Implementierung anwendbar. Es soll nun untersucht werden, in wiefern die funktionalen und nichtfunktionalen Anforderungen an das Auditing-System durch die Beispielimplementation abgedeckt werden.

Erfüllung nichtfunktionaler Kriterien

Die Proof-Of-Concept-Implementation deckt alle Komponenten des Java WS-Core, also alle Java- und webservice-basierten Teile des Globus Toolkit ab. Damit ist das nichtfunktionale *Kriterium der Vollständigkeit* (siehe Abs. 4.2) zunächst nicht erfüllt, denn die C-basierten Komponenten des Globus Toolkit (wie etwa GridFTP und GSI-SSH) werden von der Beispielimplementation nicht abgedeckt. Gleichzeitig wird aber durch die Entscheidung, zunächst eine Java-Implementierung durchzuführen, der Großteil aller anderen Dienste des Globus Toolkit unterstützt. Eine C-Implementierung für die noch übrigen Dienste wird als zukünftige Arbeit angestrebt,

stellt jedoch als solche keinen weiteren Erkenntnisgewinn dar. Sie ist im Gegenteil reine Implementationsarbeit und für das Gesamtkonzept nur von praktischem, nicht von wissenschaftlichem Interesse.

Die *Kompatibilität, Standardkonformität und Interoperabilität* der Beispielimplementation mit den übrigen eingesetzten Globus-Komponenten ist jedoch theoretisch wie praktisch voll gegeben. Wie in Kapitel 5 erläutert, baut die Implementation auf dem Globus Container auf und setzt ihre Modifikationen an der GSI so um, dass die Rückwärtskompatibilität mit nicht auditingfähigen Ressourcen voll gegeben ist. Die X.509-Zertifikatserweiterung für das Auditing wurde zu diesem Behufe als „nicht kritisch“ markiert (siehe Abs. 4.4.2); nicht auditingfähige Komponenten ignorieren sie also, anstatt die Kommunikation abzubrechen. Tests in der Grid-Umgebung der Leibniz Universität Hannover bestätigten diese Funktionalität. Auch der Auditing-Service hält sich an grid-relevante Standards, was sich inbesondere in der Verwendung von XML/WSRF für die Kommunikation mit auditingfähigen Grid-Komponenten und der Nutzung von GSI für die Sicherung des Kommunikationskanals zeigt.

Die drei Hauptanforderungen an die *Sicherheit* der Auditing-Implementierung, nämlich Authentizität, Integrität und Vertraulichkeit, werden durch die Verwendung GSI-gesicherter Kommunikation zu jedem Zeitpunkt erfüllt. Zudem ist die Auditing-Erweiterung im Proxy-Zertifikat des Nutzers nicht manipulierbar, da sie wie die restlichen Zertifikatsdaten von der nicht ohne Kenntnis des Private Key fälschbaren digitalen Signatur abhängt.

Da die Implementation auf den im Globus-Umfeld bewährten Applikationsserver „Globus Container“ zurückgreift, ist ihre *Skalierbarkeit* unmittelbar gewährleistet; alle für andere Globus-Komponenten zur Verfügung stehenden Skalierungsmöglichkeiten können auch vom Auditing-Dienst ausgeschöpft werden. Auch die als persistenter Speicher für die gesammelten Auditingdaten genutzte MySQL-Datenbank ist bei Bedarf weit skalierbar (etwa über ein Clustering oder Loadbalancing mittels mysqlproxy[2]).

Die Skalierbarkeit der Client-Komponenten ist nicht von Belang für die Untersuchung; der mögliche Flaschenhals liegt eindeutig bei dem Dienst, der Auditing-Daten entgegennimmt. Um trotz der hohen Skalierbarkeit des Globus Containers möglichen Problemen in der Skalierung des Auditing-Dienstes vorzubeugen, wurde die

[2]mysqlproxy-Projekthomepage: `https://launchpad.net/mysql-proxy`

Möglichkeit vorgesehen, mehrere URLs in der Zertifikatserweiterung unterzubringen. Somit kann bei Nichterreichbarkeit oder Überlastung eines Auditingdienstes leicht auf einen anderen ausgewichen werden.

Erfüllung funktionaler Kriterien

Die in Abschnitt 4.3 definierten funktionalen Anforderungen beziehen sich zwar ebenso wie die zuvor genannten nichtfunktionalen Anforderungen auf das Gesamtkonzept, wesentliche Teile sollen jedoch auch von der Beispielimplementation abgedeckt werden.

Die *Erstellung und Übertragung von Auditing-Einträgen* wird durch die modifizierten Komponenten des Globus WS-Core abgehandelt und den in Abschnitt 4.3 aufgezählten Anforderungen entsprechend umgesetzt. So werden die Auditingdaten nach ihrer Erhebung auf einer Grid-Ressource in einem standardisierten Format (XML) über ein im Grid universal aktzeptiertes Protokoll (WSRF) und eine GSI-gesicherte Transportverbindung verschickt.

Auch die an die *Verarbeitung und Speicherung* der Auditingdaten gestellten Anforderungen werden von der Beispielimplementation erfüllt; diese speichert alle Auditing-Tracks in einer standardisierten Form, nämlich in einer SQL-Datenbank. Der Datensicherheit wird insofern genügt, als dass diese Datenbank nicht für Dritte zugänglich ist; lediglich autorisierte Benutzer können über eine webbasierte Oberfläche zugreifen (s.u.).

Die *Missbrauchserkennung*, die in den funktionalen Anforderungen des Gesamtkonzepts gefordert wird, konnte noch keine Berücksichtigung in der Beispielimplementation finden. Das lag in dem Umstand begründet, dass für eine wirksame Missbrauchserkennung zunächst Auditingdaten in einem größeren Umfang zur Verfügung stehen mußten. Zum Zeitpunkt der Ausarbeitung der Implementation war naturgemäß keine solche Datenbasis vorhanden. Das Problem der Datenerhebung wurde schließlich – wie in Kapitel 6 erläutert – durch eine simulierte Grid-Umgebung gelöst, die als „Proof of Concept“ diente.

Wie in der Entwurfsphase geplant, wurde die Möglichkeit des Proxy-Auditing für den Nutzer optional gestaltet. Er kann also jederzeit die Kontrolle darüber ausüben, ob und wohin Auditing-Daten erhoben werden sollen. Diese *Konfigurierbarkeit* findet mithilfe der für die Beispielimplementation modifizierten Werkzeuge zur Erstellung von Grid-Proxy-Credentials statt.

Über ein webbasiertes *User-Interface* (siehe Abb. 5.2) können Auditingeinträge vom Nutzer eingesehen werden. Eine Änderung der Daten ist, sobald dise im Auditing-Dienst gespeichert sind, nicht vorgesehen (das würde die Nachvollziehbarkeit stören); mit einer konkreten Implementierung der Komponenten zur Missbrauchserkennung werden jedoch noch zusätzliche Funktionen in das User-Interface einfließen.

Performance in der Testbed-Umgebung

Ein wichtiges Kriterium für die Praxisfähigkeit der Auditing-Umgebung ist die Leistungsfähigkeit der implementierten Lösung. Zu diesem Zweck wurde die im vorigen Abschnitt 5.8 vorgestellte Testbed-Umgebung in einem prototypischen Szenario einer Performance-Messung unterzogen. Diese Messung sollte vor allem Antworten auf eine Kernfrage liefern, die laut der Entwickler des Globus Toolkit besonderes Gewicht für Nutzer und Ressourcenverwalter hat: Welche Verzögerung bei der Job Submission wird durch das Auditing induziert und ist diese Verzögerung im täglichen Betrieb tolerabel?

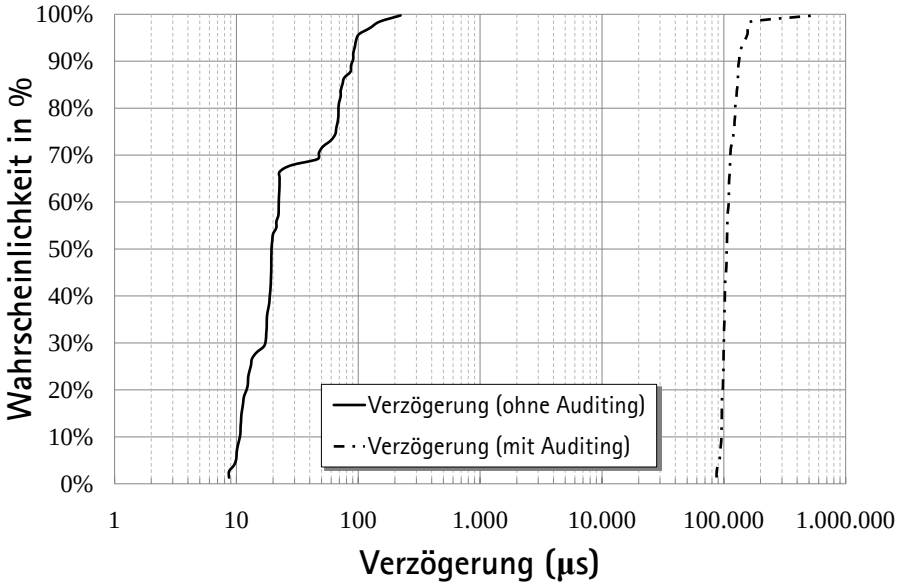


Abbildung 7.1: Performancemessung im Auditing-Testbed

Die Leistungsmessung wurde in einer virtualisierten Umgebung auf zwei virtuellen Maschinen durchgeführt, die auf demselben physikalischen Host beheimatet waren. Als Virtualisierungslösung kam VirtualBox in der zum Zeitpunkt der Messung aktuellen Version 2.2.0 zum Einsatz. Beide virtuellen Maschinen waren über

den in VirtualBox integrierten Netzwerkadapter an das Labornetz des Instituts für Rechnernetze und verteilte Systeme angeschlossen und kommunizierten untereinander mittels vom Institutsserver per DHCP zugeteilter IP-Adressen.

Diese Testumgebung eignete sich zur Durchführung von Leistungstests, weil in Produktionsumgebungen mittlerweile häufig auf Virtualisierungstechniken zurückgegriffen wird, um Komponenten mit vergleichsweise niedrigen Hardwareanforderungen (wie etwa UI-Server oder GRAMs) ressourcenschonender zu betreiben als auf eigenen physikalischen Servern. Eine Abweichung zu den Leistungsdaten in einer Wirkumgebung muß jedoch – wie bei künstlichen Performancemessungen üblich – angenommen werden.

Die durchgeführten Messungen wurden ermöglicht, indem der für das Auditing notwendige Java-WS-Core (siehe Abschnitt 5.3) weiter verändert wurde, um Timing-Informationen gewinnen zu können. So wurde direkt nach der Instanziierung der Klasse `AuditRecorder` sowie nach der Methode `AuditRecorder.sendAuditInfo()` die aktuelle Systemzeit mittels `System.nanoTime()` nanosekundengenau abgefragt. Somit kann der gesamte Zeitaufwand, der während der Job Submission für das Auditing erbracht werden muß, aufgezeichnet werden.

Zwei Durchläufe erbrachten die Messungen ohne aktiviertes Auditing (also ohne entsprechende Zertifikatserweiterung im ersten Nutzer-Proxy) sowie mit aktiviertem Auditing. Es wurden pro Durchlauf zehn Grid-Jobs vom ersten virtuellen Host, der als UI-Maschine diente, auf dem zweiten Host, der den WS-GRAM enthielt, abgegeben.

Zunächst ergab sich während der Messung die Erkenntnis, dass während jedes Grid-Jobs, der den WS-GRAM erfolgreich passiert, zehn Aufrufe der Auditing-Methoden aufgezeichnet wurden; fünf dieser Aufrufe resultierten in einer Kommunikation mit dem Auditing Service. Somit ergab sich pro Messung eine Anzahl von 50 Stichproben.

Diese Stichproben wurden nun um auffällige Ausreißer bereinigt (die jeweils auf eine vernachlässigbare Lastspitze auf den Hostsystemen hindeuteten) und in einem CDF (Abbildung 7.1) veranschaulicht.

Wie erwartet, liegt der initiale Zeitaufwand für die Instanziierung eines Objekts der Klasse AuditRecorder und die Prüfung des Proxy-Zertifikats relativ niedrig. Keine Messung lag deutlich höher als 200 μs. Wird jedoch der Auditing-Datensatz an den Auditing-Service verschickt, liegt der durch diese Kommuikation induzierte

Zeitaufwand deutlich höher (weswegen zur besseren Übersicht in Abb. 7.1 eine logarithmische Skala verwendet wurde). Der Webservice-Aufruf benötigte während der Messungen eine Zeitspanne zwischen 100 und 400 ms pro Aufruf – somit würde sich die Bearbeitung eines Grid-Jobs bei einer typischen Anzahl von 5 Auditing-Aufrufen pro Job um zwischen 500 ms und 2 s verzögern.

Diese Verzögerung liegt in der relativ aufwendigen Herstellung eines SSL-geschützten Kommunikationskanals vor jedem Auditing-Aufruf begründet und erscheint losgelöst vom Job-Kontext, in dem sie betrachtet werden muss recht hoch. Dennoch ist eine Zeitverzögerung von maximal zwei Sekunden bei einem Grid-Job, der üblicherweise Zeiträume von einigen Minuten bis mehreren Stunden zur Ausführung benötigt, ein akzeptables Resultat,denn die Job-Submission wird nicht in einem für den Nutzer merkbaren Maß verlangsamt.

Durch eine Auslagerung der Kommunikation mit dem Auditing-Dienst in einen separaten Thread könnte noch eine weitere Leistungssteigerung bewirkt werden; diese Möglichkeit ist in zukünftigen Arbeiten auszuloten.

Fazit

Die zu dem in dieser Arbeit vorgestellten Konzept entwickelte Implementation zeigt, dass die technische Umsetzbarkeit gegeben ist und das Konzept auch in der Praxis tragfähig ist. Die sich durch die Konzentration auf die Java-basierten Webservice-Komponenten des Globus Toolkit ergebenden Einschränkungen sind nicht gravierend und werden durch die Unterstützung verschiedenster Globus-Webservices aufgewogen. Bei Tests in einer virtualisierten Probeumgebung hat die Implementation unter Beweis gestellt, dass sie stabil ist und den Anforderungen einer Produktionsumgebung genügen kann.

Die Performance der Beispielimplementation ist für den Test- und Evaluationsbetrieb ausreichend und kann vor Inbetriebnahme in einer größeren Grid-Umgebung noch programmatisch verbessert werden. Die durch das Auditing induzierten Verzögerungen im Ablauf der Grid-Job-Verarbeitung wirken sich in der absoluten Mehrzahl aller Anwendungsfälle nicht gravierend auf das Gesamtsystem aus.

7.4.1 Erkennungsraten in der Simulation

Wie bereits in 6.8.1 aufgezeigt, lieferte die Missbrauchserkennung mittels eines bayesschen Klassifikators in der Simulation überzeugende Ergebnisse. Während der Prozentsatz fälschlich als mißbräuchlich einsortierter Grid-Nutzungen bei etwa 0,4% liegt, werden nur 0,7% der Gesamtmenge mißbräuchlicher Nutzungen nicht als solche erkannt. Für den Nutzer ergibt sich somit eine 99-prozentige Wahrscheinlichkeit, bei Missbrauch seiner Credentials benachrichtigt zu werden; diese Benachrichtigungen sind zudem nur sehr selten inakkurat. Bei sehr hochvolumiger Grid-Nutzung würden die regelmäßig, aber selten auftretenden irrtümlichen Warnmeldungen zwar entsprechend im Volumen ansteigen, es ist jedoch zu erwarten, dass durch eine Feedback-Schleife, also eine Reklassifikation der falsch klassifizierten Fälle aufgrund einer Nutzerrückmeldung auf Dauer eine noch höhere Erkennungsrate erzielt werden könnte.

7.5 Fazit

Im vorliegenden Kapitel fand eine kritische Würdigung der zuvor eingeführten Konzepte und ihrer exemplarischen Umsetzung anhand einer zusätzlichen Komponente für das Globus Toolkit statt. Es war zunächst zu zeigen, dass das im Kapitel 4 eingeführte Konzept einer Auditing-Infrastruktur insofern vollständig war, als dass es alle relevanten Teile des Globus Toolkit abdeckt und keine konzeptionellen Lücken vorliegen.

Ein Angreifer könnte versuchen, das Auditing-System selber zu beeinflussen. Angriffe gegen die modifizierte GSI-Bibliothek können vom Auditing-System nicht immer festgestellt und verhindert werden; um die Zuverlässigkeit des Gesamtsystems dennoch zu gewährleisten, wurde in Abschnitt 5.7 eine zusätzliche Möglichkeit aufgezeigt, über eine externe „Watchdog-Applikation“ die ordnungsgemäße Funktion auditingfähiger Komponenten zu garantieren. Mithilfe dieser Applikation ist eine sichere Funktion selbst dann gewährleistet, wenn alle Komponenten des auditingfähigen Grids als kompromittiert gelten müssen.

Die Leistungsfähigkeit der Beispielimplementation wurde in Abschnitt 7.4 untersucht. Hier wurde gezeigt, dass durch den blockierenden Versand der Auditinginformationen während des Aufbaus des Sicherheitskontextes zwar eine Verzögerung der Kommunikation induziert wird, diese jedoch im Vergleich zur gesamten Ausfüh-

rungszeit eines Grid-Jobs mit 500 bis 2000 ms sehr gering ist. Durch die Auslagerung der Kommunikation mit dem Auditingdienst in einen eigenen Thread kann diese Verzögerung noch weiter optimiert werden, so dass sie die weitere Kommunikation nicht blockiert.

Kapitel 8

Zusammenfassung und Ausblick

Das abschließende Kapitel dieser Dissertation fasst die gewonnenen Erkenntnisse und den wissenschaftlichen Beitrag dieser Arbeit in aller Kürze zusammen. Zudem zeigt der Ausblick auf zukünftige Entwicklungen und mögliche Projektträger, welche Richtung das Auditing-Projekt nach Fertigstellung dieser Dissertation nehmen kann und welche Institutionen und Initiativen hier federführend beteiligt sein könnten.

8.1 Zusammenfassung

In der vorliegenden Arbeit wurde ein Konzept zum Auditing von Grid-Proxy-Credentials vorgestellt, das zur Erkennung von Missbrauch dienen kann. Dieses Konzept gründet auf der Tatsache, daß die von jedem Grid-Nutzer zur Authentifizierung und Autorisierung auf Ressourcen verwendeten Proxy-Credentials durch diese beliebig weiterverwendet werden können. Die zu diesem Zweck gebildeten Ableitungen oder Delegationen sind nicht gegen Missbrauch gesichert und können – fallen sie in die Hände eines Angreifers – missbraucht werden, um Grid-Jobs, -Daten oder -Ressourcen zu manipulieren oder zu zerstören.

Die mangelnde Transparenz in der Verwendung von Proxy-Credentials wurde in dieser Arbeit als Hemmnis für die Grid-Nutzung identifiziert. Eine Sicherheitsanalyse der aktuellen Situation in einer exemplarischen nationalen Grid-Infrastruktur ergab zudem erhebliches Missbrauchspotential. Der Bedarf für eine Auditing-Infrastruktur wurde somit klar identifiziert und dargelegt.

Das Auditing von Grid-Proxy-Credentials findet statt, indem die Sicherheitsbibliotheken des Globus Toolkit modifiziert werden und bei jedem Authentifizierungs- und Delegationsvorgang einen Nutzungseintrag – den Audit Record – an einen zentralen Webservice senden. Dieser aggregiert die Daten und speichert sie auf eine sichere Art und Weise für die spätere Verwendung.

Eine Komponente zur automatischen Missbrauchserkennung verwendet die zuvor gesammelten Daten, um aus ihnen mit Methoden des Machine Learning mögliche Missbrauchsfälle automatisch zu ermitteln und diese an den betroffenen Nutzer zu melden. Diese automatische Erkennung erfolgt mittels bayesscher Klassifikatoren, die unter anderem die Länge und Breite einer Ableitungskette sowie die zeitliche Abfolge der auditierten Zertifikatsnutzung als Indizien für die Klassifikation nutzen.

Diese Arbeit leistet somit einen Beitrag zur Erhöhung der Sicherheit in Grid-Infrastrukturen, erhöht die Transparenz bei der Ausübung delegierter Rechte und hilft, das Vertrauen der Nutzer in das Grid zu stärken.

8.2 Ausblick

Die Methoden des wissenschaftlichen Rechnens haben sich im Bearbeitungszeitraum dieser Dissertation gewandelt. Das Grid-Computing hat sich für viele Anwendungs-

zwecke als tauglich erwiesen, wurde in anderen Bereichen aber durch die als „Cloud Computing“ bezeichnete kurzfristige und -zeitige, häufig vom Kunden selbst provisionierte Bereitstellung von IT-Infrastruktur ersetzt oder ergänzt. Nationale Grid-Initiativen wie das D-Grid in Deutschland streben zudem eine engere Zusammenarbeit im internationalen Verbund – etwa in der europäischen Gridinitative EGI – an.

Gleichwohl ist das Grid durch konstante Veränderungen und Neuentwicklungen gekennzeichnet. So erschien 2010 eine aktualisierte Version des Globus Toolkit, die – nachdem noch kurze Zeit zuvor das Webservice-Paradigma als Maß aller Dinge galt – wieder proprietäre Binärprotokolle einsetzte. Diese Entwicklung ist auch im Sommer 2011 bei vielen Grid-Ressourcen, darunter auch praktisch allen Ressourcenprovidern im D-Grid, noch nicht flächendeckend eingeführt worden. Diese zögern noch, weil zentrale Fragen zur Migration von Diensten und Ressourcen nicht adäquat beantwortet werden können.

Die Globus Alliance hat in Anerkenntnis des nicht zu ignorierenden Siegeszuges cloudbasierter Dienste und Anwendungen einen eigenen Cloud-Dienst namens „Globus Online“ ins Leben gerufen, der den Datentransfer vom Client-Rechner des Wissenschaftlers ins Grid deutlich vereinfachen soll. Dazu wird aber ein Proxy-Credential benötigt, das der Anwender „in die Cloud“ hochlädt und das dort vollkommen unkontrolliert von den Globus-Online-Diensten verwendet werden kann. Fragen der Haftung, des Datenschutzes und der Vertraulichkeit müssen hier ausgeklammert werden – die Nutzung von „Globus Online“ kommt einem totalen Kontrollverlust durch den Nutzer gleich.

Wissenschaftler der Universität Dortmund arbeiten hingegen an einer Schnittstelle zwischen EC2-kompatiblen Clouds und dem D-Grid, um einen Brückenschlag zwischen Grid-Infrastrukturen und Cloud-Middlewares zu ermöglichen [SE10].

Auch in diesen Cloud-Grid-Hybridszenarios kann das Proxy-Auditing wertvolle Dienste leisten. Gelingt es, Anbieter von der Wichtigkeit einer kontrollierten Nutzung der Delegationsmöglichkeiten von Grid-Proxy-Credentials zu überzeugen und die hier vorgestellte Lösung zur Überwachung von Credentials zu etablieren, werden Sicherheitsfragen des Cloud Computing genauso gelöst, wie sie für das Grid Computing gelöst werden können.

8.2.1 Zukünftige Erweiterungen

Mit dem hier vorliegenden Konzept wurde die konzeptionelle und technische Basis für eine gridweite generische Auditing-Infrastruktur gelegt. Zwar wurde in dieser Dissertation ein Anwendungsfall – nämlich die Auditierung von Grid-Proxy-Credentials – besonders detailliert betrachtet, dennoch kann das Konzept des Auditings mittels eines Globus-Webservice auch für verschiedene andere Anwendungsfälle genutzt werden.

So könnte die Ausführung und Manipulation von Grid-Jobs auditiert werden, um die Ausführung bestimmter Programme durch unbefugte Dritte zu erkennen. Ebenso könnte die Auditierung über ihre eigentliche Intention der Angriffserkennung auch zum Zwecke der Repudiation, also der wissenschaftlichen Nachvollziehbarkeit genutzt werden. Beim Auditing von Datentransfers und -zugriffen ließe sich ein Dateicontainerformat erdenken, das bei jedem Kopiervorgang im Grid durch das Auditing nachverfolgt würde. So könnte nicht nur die unbefugte Vervielfältigung von Dateien zurückverfolgt werden, sondern auch aufgrund der Auditingdaten ermittelte häufige Transferwege im Grid entsprechend optimiert werden.

Durch die flexible Architektur der Auditing-Infrastruktur sind derartige Erweiterungen leicht zu implementieren; sie können auf die bereits getesteten und erprobten Konzepte aus dieser Dissertation zurückgreifen. Auch eine Übertragung in Cloud- oder Web-Infrastrukturen ist denkbar, denn das Auditing von X.509-Credentials ist nicht nur mit Proxy-, sondern auch mit entsprechend modifizierten End-Entity-Credentials möglich.

Eine im Zuge des vorliegenden Projekts intensiv diskutierte Frage ist die Rückwärtskompatibilität der auditingfähigen Proxy-Credentials. Während in der Anfangsphase des Projekts auf maximale Kompatibilität mit den vorhandenen, nicht auditingfähigen Grid-Ressourcen begründet Wert gelegt wurde, könnte bei ausreichender Verbreitung diese Kompatibilität eingeschränkt werden, um mehr Sicherheit zu gewinnen. Diese Paradigmenverschiebung äußert sich in der Änderung der Auditing-Erweiterung im X.509-Zertifikat von „non-critical“ zu „critical“. Ist eine Extension als „critical“ markiert, so müssen zertifikatsverarbeitende Komponenten, die die Extension nicht kennen (also nicht auditinfähige Grid-Ressourcen) das betreffende Zertifikat ablehnen und die Kommunikation beenden. Effektiv heißt das, daß eine Nutzung nicht auditingfähiger Grid-Ressourcen mit einem auditingfähigen Proxy-Credential nicht mehr möglich wäre. Ein Nutzer könnte sich mithilfe dieses

Ausschlusses bewußt dafür entscheiden, keine Delegationen an Grid-Dienste abzugeben, die nicht auditingfähig sind und somit keine Möglichkeit der Nachverfolgung seiner Proxy-Credentials erlauben. Um eine solche Option anzubieten, müsste jedoch eine kritische Masse an auditingfähigen Ressourcen erreicht werden, die derzeit nicht vorhanden ist.

8.2.2 Zukunft des Projekts

Das in dieser Dissertation präsentierte Konzept wurde auf verschiedenen wissenschaftlichen Konferenzen und Symposien vorgestellt und in verschiedene Forschungsprojekte eingebunden. Neben der engen Verzahnung mit dem D-Grid ist hier zunächst die Aufnahme in das „Globus Incubator“-Programm zu nennen, die im Jahr 2009 erfolgte. Das Incubator-Programm besteht aus Erweiterungen und Diensten für das Globus Toolkit, die nicht in die Kerndistribution aufgenommen, wohl aber von Tutoren der Globus Alliance betreut werden. Der Diskurs mit den Entwicklern des Globus Toolkit brachte wertvolle Anregungen und Kritik in das Projekt ein. Im Zuge der Umstellungen während der Test- und Einführungsphase von Globus 5 wurde das Incubator-Projekt jedoch eingefroren; sein Status ist heute ungeklärt.

Die nationale Grid-Infrastruktur Großbritanniens, der „NGS“ (National Grid Service), hat basierend auf dem hier vorgestellten Konzept und der als Open-Source-Programm verfügbaren Referenzimplementation eine eigene Auditing-Infrastruktur aufgebaut und für diese Infrastruktur ein Projekt beim zuständigen Mittelgeber beantragt. Dieses Projekt im europäischen Kontext wird die Verbreitung und Relevanz des Auditing noch zusätzlich erhöhen.

Generell ist zum Zeitpunkt der Fertigstellung dieser Dissertation im Sommer 2011 unklar, wie die Sicherheitsinfrastruktur des Grid in Zukunft aussehen wird. Erste Diskussionen rund um die Nachfolgeversion des Globus Toolkit 5 lassen vermuten, dass die GSI in der jetzigen Form von einer verschlankten Variante abgelöst wird; Proxy-Zertifikate werden nach Aussage der Globus-Entwickler jedoch auch in zukünftigen Versionen das Mittel der Wahl zur Authentifizierung und Autorisierung sein.

Literaturverzeichnis

[AAB+05] Antonioletti, Mario ; Atkinson, Malcolm ; Baxter, Rob ; Borley, Andrew ; Chue Hong, Neil P. ; Collins, Brian ; Hardman, Neil ; Hume, Alastair C. ; Knox, Alan ; Jackson, Mike ; Krause, Amy ; Laws, Simon ; Magowan, James ; Paton, Norman W. ; Pearson, Dave ; Sugden, Tom ; Watson, Paul ; Westhead, Martin: The design and implementation of Grid database services in OGSA-DAI. In: *Concurrency and Computation: Practice and Experience* 17 (2005), Nr. 2-4, S. 357–376. http://dx.doi.org/10.1002/cpe.939. – DOI 10.1002/cpe.939. – ISSN 1532–0634

[ALMS89] Ammann, Thomas ; Lehnhardt, Matthias ; Meissner, Gerd ; Stahl, Stephan: *Hacker für Moskau. Deutsche Computer- Spione im Dienst des KGB*. Wunderlich Verlag, 1989. – ISBN 3–8052–0490–6

[BB05] Baker, Mark A. ; Boakes, Richard J.: Semantic Logging using the Resource Description Framework. In: *CCGrid: Special Session for Work-in-Progress Papers on Novel Grid Technologies*, 2005

[BM94] Bellovin, S.M. ; Merritt, M.: An attack on the Interlock Protocol when used for authentication. In: *IEEE Transactions on Information Theory* 40 (1994), S. 273 – 275. http://dx.doi.org/10.1109/18.272497. – DOI 10.1109/18.272497

[Bun] Bundesamt für Sicherheit in der Informationstechnik: *IT-Grundschutz-Kataloge*. [Online]. http://www.bsi.bund.de/gshb/index.htm,

[CFF+04] Czajkowski, Karl ; Ferguson, Donald F. ; Foster, Ian ; Frey, Jeffrey ; Graham, Steve ; Sedukhin, Igor ; Snelling, David ; Tu-

ECKE, Steve ; VAMBENEPE, William: *The WS-Resource Framework.* [Online]. http://www.globus.org/wsrf/specs/ws-wsrf.pdf, 2004

[DH76a] DIFFIE, Whitfield ; HELLMAN, Martin E.: Multiuser cryptographic techniques. In: *AFIPS '76: Proceedings of the June 7-10, 1976, national computer conference and exposition.* New York, NY, USA : ACM, 1976, S. 109–112

[DH76b] DIFFIE, Whitfield ; HELLMAN, Martin E.: New Directions in Cryptography. In: *IEEE Transactions on Information Theory* IT-22 (1976), Nr. 6, 644–654. http://citeseer.ist.psu.edu/diffie76new.html

[EGK+07] ELLERT, M. ; GRONAGER, M. ; KONSTANTINOV, A. ; KONYA, B. ; LINDEMANN, J. ; LIVENSON, I. ; NIELSEN, J. ; NIINIMAKI, M. ; SMIRNOVA, O. ; WAANANEN, A.: Advanced Resource Connector middleware for lightweight computational Grids. In: *Future Generation Computer Systems* 23 (2007), Februar, Nr. 2, S. 219–240. http://dx.doi.org/10.1016/j.future.2006.05.008. – DOI 10.1016/j.future.2006.05.008. – ISSN 0167–739–X

[EJ01] EASTLAKE, D. ; JONES, P.: *RFC 3174 - US Secure Hash Algorithm 1 (SHA1).* [Online] http://www.faqs.org/rfcs/rfc3174.html, September 2001

[ElG85] ELGAMAL, Taher: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Version: 1985. http://dx.doi.org/10.1007/3-540-39568-7_2. In: BLAKLEY, George (Hrsg.) ; CHAUM, David (Hrsg.): *Advances in Cryptology* Bd. 196. Springer Berlin / Heidelberg, 1985. – DOI 10.1007/3–540–39568–7_2, S. 10–18

[EUG06] EUGRIDPMA: *European Policy Management Authority for Grid Authentication.* [Online]. http://eugridpma.org/, 2006

[EUG08] EUGRIDPMA: *OID for Proxy Delegation Tracing.* [Online]. http://www.eugridpma.org/documentation/OIDProxyDelegationTracing.pdf, 2008

[Eur11] EUROPEAN GRID INITIATIVE: *EGI: About Us.* [Online]. http://www.egi.eu/about/, June 2011

[Fal06] Falkner, Jürgen: Auditing und Tracking. In: *Deliverable 3 - D-Grid Security-Workshop, Version 1.0*, 2006, S. 183 – 202

[FH02] Farrell, S. ; Housley, R.: *RFC 3281: An Internet Attribute Certificate Profile for Authorization*. [Online]. `http://www.ietf.org/rfc/rfc3281.txt`, April 2002

[Fie09] Fieseler, Thomas: *Betriebskonzept für die D-Grid Infrastruktur*. [Online]. `http://www.d-grid.de/fileadmin/user_upload/documents/Kern-D-Grid/Betriebskonzept/D-Grid-Betriebskonzept.pdf`, Dezember 2009

[FK03] Foster, Ian ; Kesselman, Carl: *The Grid 2: Blueprint for a New Computing Infrastructure*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2003. – ISBN 1–5586–0933–4

[FKT01] Foster, Ian ; Kesselman, Carl ; Tuecke, Steven: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. In: *Euro-Par '01: Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*. London, UK : Springer-Verlag, 2001. – ISBN 3–540–42495–4, S. 1–4

[FKTT98] Foster, I. ; Kesselman, C. ; Tsudik, G. ; Tuecke, S.: A Security Architecture for Computational Grids. In: *Proceedings of the 5th ACM Conference on Computer and Communications Security*. New York, NY : ACM Press, 1998, S. 83–91

[FMC11] Falliere, Nicolas ; Murchu, Liam O. ; Chien, Eric: *W32.Stuxnet Dossier*. [Online]. `http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf`, Februar 2011

[Fos02] Foster, Ian: What is the Grid? - a three point checklist. In: *GRIDtoday* (2002), Juli, Nr. 6

[Fos05] Foster, Ian: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, Hai (Hrsg.) ; Reed, Daniel A. (Hrsg.) ; Jiang, Wenbin (Hrsg.): *NPC* Bd. 3779, Springer, 2005 (Lecture Notes in Computer Science). – ISBN 3–540–29810–X, S. 2–13

[FSF06] Friese, Thomas ; Smith, Matthew ; Freisleben, Bernd: GDT: A Toolkit for Grid Service Development. In: *In: Proc. of the 3rd International Conference on Grid Service Engineering and Management*, 2006, S. 131–148

[Fuh04] Fuhrmann, Patrick: dCache: the commodity cache. In: *In Twelfth NASA Goddard and Twenty First IEEE Conference on Mass Storage Systems and Technologies*, 2004

[GHM06] Graham, Steve ; Hull, David ; Murray, Bryan: *Web Services Base Notification 1.3*. [Online]. `http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf`, Oktober 2006

[GKG09] Groeper, R. ; Kunz, C. ; Grimm, C.: Connecting OGC web services and the Grid using Globus Toolkit 4 and OGSA-DAI. In: *Grid Computing, 2009 10th IEEE/ACM International Conference on*, 2009, S. 66 –73

[Gra02] Graham, Paul: *A plan for Spam*. [Online]. `http://www.paulgraham.com/spam.html`, August 2002

[He08] He, Lian-yun: Application of Bayesian Network in Power Grid Fault Diagnosis. In: *Natural Computation, 2008. ICNC '08. Fourth International Conference on* 1 (2008), Oktober, S. 61–64. `http://dx.doi.org/10.1109/ICNC.2008.425`. – DOI 10.1109/ICNC.2008.425

[HFH+09] Hall, Mark ; Frank, Eibe ; Holmes, Geoffrey ; Pfahringer, Bernhard ; Reutemann, Peter ; Witten, Ian H.: The WEKA data mining software: an update. In: *SIGKDD Explor. Newsl.* 11 (2009), November, S. 10–18. `http://dx.doi.org/10.1145/1656274.1656278`. – DOI 10.1145/1656274.1656278. – ISSN 1931–0145

[HgE+10] Hommel, Wolfgang ; gentschen Felde, Nils ; Eye, Felix von ; Kohlrausch, Jan ; Szongott, Christian: *Architekturkonzept für ein Grid-basiertes IDS*. [Online]: `http://www.grid-ids.de/documents/GIDS_MS16-1.pdf`, Oktober 2010

[Hyp11] Hypponen, Mikko: *Found: Bitcoin mining bot that is controlled via Twitter*. [Online]. http://www.f-secure.com/weblog/archives/00002207.html, August 2011

[Int08a] International Standardization Organization: *ISO/IEC 27001:2005: Information technology - Security techniques - Information security management systems - Requirements*. Oktober 2008

[Int08b] International Standardization Organization: *ISO/IEC 27002:2005: Information technology -Security techniques - Code of practice for information security management*. April 2008

[ITU02] ITU-T Study Group 17: *Recommendation X.690 - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. Online: http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf, 07 2002

[ITU05] ITU-T Study Group 17: *Recommendation X.509 - The Directory: Public-key and attribute certificate frameworks*. [Online]. http://www.itu.int/rec/T-REC-X.509/en, August 2005

[ITU08a] ITU-T Study Group 17: *Recommendation X.500: The Directory: Overview of concepts, models and services*. [Online]. http://www.itu.int/rec/T-REC-X.500-200811-I/en, November 2008

[ITU08b] ITU-T Study Group 17: *Recommendation X.660 - Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the International Object Identifier tree*. [Online]. http://www.itu.int/rec/T-REC-X.660-200808-I/en, August 2008

[ITU08c] ITU-T Study Group 17: *Recommendation X.680 - Abstract Syntax Notation One (ASN.1): Specification of basic notation*. [Online]. http://www.itu.int/rec/T-REC-X.680-200811-I/en, November 2008

[ITU08d] ITU-T Study Group 17: *Recommendation X.681 - Abstract Syntax Notation One (ASN.1): Information object specification*. [Online]. http://www.itu.int/rec/T-REC-X.681-200811-I/en, November 2008

[KBB+09] Krefting, Dagmar ; Bart, J. ; Beronov, K. ; Dzhimova, O. ; Falkner, J. ; Hartung, M. ; Hoheisel, A. ; Knoch, T.A. ; Lingner, T. ; Mohammed, Y. ; Peter, K. ; Rahm, E. ; Sax, U. ; Sommerfeld, D. ; Steinke, T. ; Tolxdorff, T. ; Vossberg, M. ; Viezens, F. ; Weisbecker, A.: MediGRID: Towards a user friendly secured grid infrastructure. In: *Future Generation Computer Systems* 25 (2009), Nr. 3, S. 326 – 336. – ISSN 0167–739X

[KE08] Kunz, Christopher ; Esser, Stefan: *PHP-Sicherheit*. 3. erweiterte Auflage. dpunkt.Verlag Heidelberg, 2008. – ISBN 978–3898645355

[KSWG09] Kunz, C. ; Szongott, C. ; Wiebelitz, J. ; Grimm, C.: Design and Implementation of a Grid Proxy Auditing Infrastructure. In: *eScience 2009, 5th IEEE International Conference on*, 2009

[KTRS11] Kunz, Christopher ; Tahmasebi, Nina ; Risse, Thomas ; Smith, Matthew: Detecting Credential Abuse in the Grid Using Bayesian Networks. In: *Grid Computing (GRID), 2011 12th IEEE/ACM International Conference on*, 2011. – ISSN 1550–5510, S. 114 –120

[Kul09] Kulke, Ulli: *Die Tricks der Forscher beim Klimawandel*. [Online]. `http://www.welt.de/wissenschaft/article5294872/Die-Tricks-der-Forscher-beim-Klimawandel.html`, November 2009

[KWPG09a] Kunz, C. ; Wiebelitz, J. ; Piger, S. ; Grimm, C.: A Concept for Grid Credential Lifecycle Management and Heuristic Credential Abuse Detection. In: *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, 2009, S. 505–510

[KWPG09b] Kunz, C. ; Wiebelitz, J. ; Piger, S. ; Grimm, C.: A Concept for Grid Credential Lifecycle Management and Heuristic Credential Abuse Detection. In: *Parallel and Distributed Computing, 2009. ISPDC '09. Eighth International Symposium on*, 2009, S. 245 –248

[KWS10] Kunz, Christopher ; Wiebelitz, Jan ; Smith, Matthew: An attack-resilient Grid auditing infrastructure. In: *Wireless Communications,*

Networking and Information Security (WCNIS), 2010 IEEE International Conference on, 2010, S. 635 – 639

[LFS+06] Lang, Bo ; Foster, Ian ; Siebenlist, Frank ; Ananthakrishnan, Rachana ; Freeman, Tim: A Multipolicy Authorization Framework for Grid Security. In: *Proceedings of the Fifth IEEE Symposium on Network Computing and Application*, 2006, S. 269–272

[LHP+04] Laure, E ; Hemmer, F ; Prelz, F ; Beco, S ; Fisher, S ; Livny, M ; Guy, L ; Barroso, M ; Buncic, P ; Kunszt, Peter Z. ; Di Meglio, A ; Aimar, A ; Edlund, A ; Groep, D ; Pacini, F ; Sgaravatto, M ; Mulmo, O: Middleware for the next generation Grid infrastructure. (2004), Nr. EGEE-PUB-2004-002, S. 4 pp.

[LMM05] Luna, Jesus ; Medina, Manel ; Manso, Oscar: Towards a Unified Authentication and Authorization Infrastructure for Grid Services: Implementing an Enhanced OCSP Service Provider into GT4. In: Chadwick, David W. (Hrsg.) ; Zhao, Gansen (Hrsg.): *EuroPKI* Bd. 3545, Springer, 2005 (Lecture Notes in Computer Science). – ISBN 3–540–28062–6, S. 36–54

[LMM07] Luna, Jesus ; Medina, Manel ; Manso, Oscar: Using OGRO and CertiVeR to improve OCSP validation for Grids. In: *The Journal of Supercomputing* 42 (2007), Nr. 3, S. 253–266

[LS94] Langley, Pat ; Sage, Stephanie: Induction of Selective Bayesian Classifiers. In: *Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 1994, S. 399–406

[Lun08] Luna, Jesus: *A Dynamic Validation Infrastructure for Interoperable Grid Services*, Universidad Politécnica de Cataluña, Diss., February 2008

[MAM+99] Myers, M. ; Ankney, R. ; Malpani, A. ; Galperin, S. ; Adams, C.: *RFC 2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. [Online]. `http://www.ietf.org/rfc/rfc2560.txt`, Juni 1999

[MT07] Meints, M. ; Thomsen, S.: Protokollierung in Sicherheitsstandards. In: *Datenschutz und Datensicherheit, Vieweg Verlag* 31 (2007), S. 749–751

[Mur98] Murphy, Kevin: A Brief Introduction to Graphical Models and Bayesian Networks. (1998). http://www.cs.ubc.ca/~murphyk/Bayes/bayes.html

[NGG+07] Nadalin, A. ; Goodner, M. ; Gudgin, M. ; Barbir, A ; Granqvist, H.: *WS-SecureConversation 1.3.* [Online]. http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.pdf, März 2007

[NKMHB06] Nadalin, A. ; Kaler, C. ; Monzillo, R. ; Hallam-Baker, P.: *Web Services Security: SOAP Message Security.* [Online]. http://docs.oasis-open.org/wss/v1.1/, February 2006

[NTW01] Novotny, J. ; Tuecke, S. ; Welch, V.: An Online Credential Repository for the Grid: MyProxy. In: *HPDC '01: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01).* Washington, DC, USA : IEEE Computer Society, 2001, S. 104

[NYHR05] Neuman, C. ; Yu, T. ; Hartman, S. ; Raeburn, K: *RFC4120 - The Kerberos Network Authentication Service (V5).* [Online]. http://tools.ietf.org/html/rfc4120, July 2005

[Pea96] Pearl, Judea: Bayesian Networks. In: *MIT Encyclopedia of the Cognitive Sciences* (1996). http://preprints.stat.ucla.edu/223/223.pdf

[PGGK08] Piger, Stefan ; Grimm, Christian ; Groeper, Ralf ; Kunz, Christopher: A Comprehensive Approach to Self-Restricted Delegation of Rights in Grids. In: *Cluster Computing and the Grid, IEEE International Symposium on.* Los Alamitos, CA, USA : IEEE Computer Society, 2008, S. 114–121

[Pig08] Piger, Stefan: *Nutzerdefinierte Restriktion delegierter Privilegien im Grid-Computing*, Gottfried Wilhelm Leibniz Universität Hannover, Diss., 2008

[PKGG08] Piger, S. ; Kunz, C. ; Grimm, C. ; Groeper, R.: Enhancing Security in Grids through Self-Restricted Delegation of Rights with User-based Policies. In: *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN2008)*, 2008

[PWF+02] Pearlman, L. ; Welch, V. ; Foster, I. ; Kesselman, C. ; Tuecke, S.: A Community Authorization Service for Group Collaboration. In: *Policies for Distributed Systems and Networks, IEEE International Workshop on* 0 (2002), S. 0050. ISBN 0–7695–1611–4

[Rom99] Romberg, Mathilde: The UNICORE Architecture: Seamless Access to Distributed Resources. In: *High-Performance Distributed Computing, International Symposium on* (1999), S. 44. – ISSN 1082–8907

[RS84] Rivest, Ronald L. ; Shamir, Adi: How to expose an eavesdropper. In: *Communications of the ACM* 27 (1984), Nr. 4, S. 393 – 394. `http://dx.doi.org/10.1145/358027.358053`. – DOI 10.1145/358027.358053. – ISSN 0001–0782

[RSA77] Rivest, Ronald L. ; Shamir, Adi ; Adleman, Len: On Digital Signatures and Public-Key Cryptosystems. (1977), April

[RSA78] Rivest, R. L. ; Shamir, A. ; Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. In: *Commun. ACM* 21 (1978), Nr. 2, S. 120–126. – ISSN 0001–0782

[SC06] Sotomayor, Borja ; Childers, Lisa: *Globus Toolkit 4 - Programming Java Services*. Elsevier, 2006

[SDHH98] Sahami, Mehran ; Dumais, Susan ; Heckerman, David ; Horvitz, Eric: A Bayesian Approach to Filtering Junk E-Mail. (1998). `http://robotics.stanford.edu/users/sahami/papers-dir/spam.ps`

[SE10] Schwiegelshohn, Uwe ; Eberhart, Andreas: *Erweiterung der D-Grid Basis für die kommerzielle Nutzung.* [Online].

http://www.irf.tu-dortmund.de/cms/de/IT/Projekte/D-Grid_IaaS/Vorhabenbeschreibung.pdf, März 2010

[SPUP02] Schein, Andrew I. ; Popescul, Alexandrin ; Ungar, Lyle H. ; Pennock, David M.: Methods and Metrics for Cold-Start Recommendations. In: *Proceedings Of The 25th Annual International ACM SIGIR Conference On Research And Development In Information Retrieval*, 2002, S. 253–260

[SSH+09] Smith, M. ; Schwarzer, F. ; Harbach, M. ; Noll, T. ; Freisleben, B.: A Streaming Intrusion Detection System for Grid Computing Environments. In: *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, 2009, S. 44 –51

[Str08] Stratmann, Mirko: *Entwurf und Implementierung eines Auditing-Service für Grid-Jobs*. Masterarbeit an der Gottfried Wilhelm Leibniz Universität Hannover, 2008

[Szo09] Szongott, Christian: *Webservice-basiertes Auditing für Grid Proxy Credentials*. Masterarbeit an der Gottfried Wilhelm Leibniz Universität Hannover, Mai 2009

[The07] The Globus Alliance: *GRAM Audit Logging*. [Online]. http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS_GRAM_Audit_Logging.html, 2007

[TWE+04] Tuecke, Steve ; Welch, Von ; Engert, Douglas ; Pearlman, Laura ; Thompson, Mary: *RFC 3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. [Online]. http://www.ietf.org/rfc/rfc3820.txt, Juni 2004

[Ver90] Verschiedene Autoren: *Bundesdatenschutzgesetz*. [Online]. http://www.gesetze-im-internet.de/bdsg_1990/index.html, 1990

[Ver01] Verschiedene Autoren: *Announcing the Advanced Encryption Standard (AES)*. [Online]. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, November 2001

[VG07] Volpato, Gian L. ; Grimm, Christian: *Recommendations for Static Firewall Configuration in D-Grid.* [Online]. `http://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-5/FG3-5_Recommendations_Static_Firewall.pdf`, 2007

[WBKS10] Wiebelitz, Jan ; Brenner, Michael ; Kunz, Christopher ; Smith, Matthew: Early defense: enabling attribute-based authorization in Grid firewalls. In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing.* New York, NY, USA : ACM, 2010 (HPDC '10). – ISBN 978–1–60558–942–8, 336–339

[WKPG09] Wiebelitz, J. ; Kunz, C. ; Piger, S. ; Grimm, C.: TCP-AuthN: An Approach to Dynamic Firewall Operation in Grid Environments. In: *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, 2009, S. 481 –486

[WPKG09] Wiebelitz, Jan ; Piger, Stefan ; Kunz, Christopher ; Grimm, Christian: Transparent Identity-based Firewall Transition for eScience. In: *Proceedings of the 2009 Fifth IEEE International Conference on e-Science*, 2009

[Zei06] Zeilenga, K.: *RFC 4510 - Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map.* [Online]. `http://tools.ietf.org/html/rfc4510`, Juni 2006

[Zha08] Zhang, Jun: *Storm Worm & Botnet Analysis.* [Online]. `http://securitylabs.websense.com/content/Assets/Storm_Worm_Botnet_Analysis_-_June_2008.pdf`, Juni 2008

Tabellarischer Lebenslauf

Christopher Kunz

Persönliche Daten

22. Oktober 1979	geboren in Gütersloh als Sohn von Günter Kunz und Ute Kunz (geb. Zimmermann).

Schulische und akademische Ausbildung

1986–1990	Grundschule Parkschule Rheda-Wiedenbrück.
1990–1999	Ratsgymnasium Rheda-Wiedenbrück.
2000–2005	Gottfried Wilhelm Leibniz Universität Hannover, Studium der angewandten Informatik (später: Informatik), Abschluß Bachelor of Science.
2006–2007	Gottfried Wilhelm Leibniz Universität Hannover, Studium der Informatik , Abschluß Master of Science.
August 2007	Abschluss mit der Masterprüfung.

Wissenschaftlicher Werdegang

seit September 2007	Gottfried Wilhelm Leibniz Universität Hannover, Lehrgebiet Rechnernetze, Wissenschaftlicher Mitarbeiter.
2007–2008	Gottfried Wilhelm Leibniz Universität Hannover, Lehrgebiet Rechnernetze, Mitarbeit im BMBF-Projekt *Geodaten-Infrastruktur-Grid* (GDI-Grid) Arbeitspaket 2: „Authentifizierungs- und Autorisierungsinfrastrukturen“(AAI), Datenmanagement, VO-Management
2008–2010	Gottfried Wilhelm Leibniz Universität Hannover, Lehrgebiet Rechnernetze, Mitarbeit im BMBF-Projekt *Geodaten-Infrastruktur-Grid* (GDI-Grid) Arbeitspaket 1: „Koordination und Projektmanagement“

seit Januar 2011	Gottfried Wilhelm Leibniz Universität Hannover, Lehrgebiet Rechnernetze, Mitarbeit im BMBF-Projekt *D-Grid / NGI-DE*, Arbeitspaket „Sicherheit“

Wissenschaftliche Veröffentlichungen als Erstautor

[KWPG09a]	Kunz, C. ; Wiebelitz, J. ; Piger, S. ; Grimm, C.: A Concept for Grid Credential Lifecycle Management and Heuristic Credential Abuse Detection. In: *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, 2009, S. 505–510
[KWPG09b]	Kunz, C. ; Wiebelitz, J. ; Piger, S. ; Grimm, C.: A Concept for Grid Credential Lifecycle Management and Heuristic Credential Abuse Detection. In: *Parallel and Distributed Computing, 2009. ISPDC '09. Eighth International Symposium on*, 2009, S. 245 –248
[KSWG09]	Kunz, C. ; Szongott, C. ; Wiebelitz, J. ; Grimm, C.: Design and Implementation of a Grid Proxy Auditing Infrastructure. In: *eScience 2009, 5th IEEE International Conference on*, 2009
[KWS10]	Kunz, Christopher ; Wiebelitz, Jan ; Smith, Matthew: An attack-resilient Grid auditing infrastructure. In: *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, 2010, S. 635 –639
[KTRS11]	Kunz, Christopher ; Tahmasebi, Nina ; Risse, Thomas; Smith, Matthew: Detecting Credential Abuse in the Grid Using Bayesian Networks. In: *Grid Computing (GRID), 2011 12th IEEE/ACM International Conference on*, 2011. – ISSN 1550–5510, S. 114 –120

Wissenschaftliche Veröffentlichungen als Koautor

[PGGK08]	Piger, Stefan ; Grimm, Christian ; Groeper, Ralf ; Kunz,

Christopher: A Comprehensive Approach to Self-Restricted Delegation of Rights in Grids. In: *Cluster Computing and the Grid, IEEE International Symposium on* Bd. 0. Los Alamitos, CA, USA : IEEE Computer Society, 2008, S. 114–121

[PKGG08] Piger, S. ; Kunz, C. ; Grimm, C. ; Groeper, R.: Enhancing Security in Grids through Self-Restricted Delegation of Rights with User-based Policies. In: *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN2008)*, 2008

[GKG09] Groeper, R. ; Kunz, C. ; Grimm, C.: Connecting OGC web services and the Grid using Globus Toolkit 4 and OGSA-DAI. In: *Grid Computing, 2009 10th IEEE/ACM International Conference on*, 2009, S. 66 –73

[WPKG09] Wiebelitz, Jan ; Piger, Stefan ; Kunz, Christopher ; Grimm, Christian: Transparent Identity-based Firewall Transition for eScience. In: *Proceedings of the 2009 Fifth IEEE International Conference on e-Science*, 2009

[WKPG09] Wiebelitz, J. ; Kunz, C. ; Piger, S. ; Grimm, C.: TCP-AuthN: An Approach to Dynamic Firewall Operation in Grid Environments. In: *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, 2009, S. 481 –486

[WBKS10] Wiebelitz, Jan ; Brenner, Michael ; Kunz, Christopher ; Smith, Matthew: Early Defense - Enabling Attribute-Based Authorization in Grid Firewalls, 2010, S. 336–339

Andere Veröffentlichungen

[KE08] Kunz, Christopher ; Esser, Stefan: *PHP-Sicherheit*. 3. erweiterte Auflage. dpunkt.Verlag Heidelberg, 2008