A HOLISTIC APPROACH TO NETWORK SECURITY IN OGSA-BASED GRID SYSTEMS

By

Demetrios Loutsios

A HOLISTIC APPROACH TO NETWORK SECURITY IN OGSA-BASED GRID SYSTEMS

By

Demetrios Loutsios

Dissertation

submitted in fulfilment
of the requirements
for the degree

Magister Technologiae

in

Information Technology

in the

**School of ICT: Faculty of Engineering,
the Built Environment, and
Information Technology**

of the

Nelson Mandela Metropolitan University

Supervisor: Dr. Maree Pather

December 2006

DECLARATION

I Demetrios Loutsios, hereby declare that:

- The work in this dissertation is my own work.
- All sources used r referred to have been documented and recognized.
- This dissertation has not previously been submitted in full or partial fulfilment of the requirements for an equivalent or higher qualification at any other recognized educational institutions

_____

Demetrios Loutsios

A HOLISTIC APPROACH TO NETWORK SECURITY IN OGSA-BASED GRID SYSTEMS

Abstract

Grid computing technologies facilitate complex scientific collaborations between globally dispersed parties, which make use of heterogeneous technologies and computing systems. However, in recent years the commercial sector has developed a growing interest in Grid technologies. Prominent Grid researchers have predicted Grids will grow into the commercial mainstream, even though its origins were in scientific research. This is much the same way as the Internet started as a vehicle for research collaboration between universities and government institutions, and grew into a technology with large commercial applications.

Grids facilitate complex trust relationships between globally dispersed business partners, research groups, and non-profit organizations. Almost any dispersed "virtual organization" willing to share computing resources can make use of Grid technologies.

Grid computing facilitates the networking of shared services; the inter-connection of a potentially unlimited number of computing resources within a "Grid" is possible. Grid technologies leverage a range of open standards and technologies to provide interoperability between heterogeneous computing systems. Newer Grids build on key capabilities of Web-Service technologies to provide easy and dynamic publishing and discovery of Grid resources.

Due to the inter-organisational nature of Grid systems, there is a need to provide adequate security to Grid users and to Grid resources. This research proposes a framework, using a specific brokered pattern, which addresses several common Grid security challenges, which include:

- Providing secure and consistent cross-site Authentication and Authorization;
- Single-sign on capabilities to Grid users;

- Underlying platform and runtime security, and;
- Grid network communications and messaging security.

These Grid security challenges can be viewed as comprising two (proposed) logical layers of a Grid. These layers are: *a Common Grid Layer* (higher level Grid interactions), and *a Local Resource Layer* (Lower level technology security concerns). This research is concerned with providing a generic and holistic security framework to secure both layers. This research makes extensive use of STRIDE - an acronym for Microsoft approach to addressing security threats - as part of a holistic Grid security framework.

STRIDE and key Grid related standards, such as Open Grid Service Architecture (OGSA), Web-Service Resource Framework (WS-RF), and the Globus Toolkit are used to formulate the proposed framework.

Acknowledgements

My sincere thanks are due to:

- My supervisor, Dr. Maree Pather for his guidance and commitment to my research, all the stimulating conversations over the years, and his valuable guidance in matters outside my research.
- My family, in particular my father Chris Loutsios for many years of encouragement, and all the sacrifices that he has made for me to obtain an education.
- James and Riekert for their friendship and encouragement.
- Most importantly, God for giving me the strength and ability to produce this dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Grid computing has gained in popularity and application in recent years. There has been a growing trend towards interconnected systems both within and across enterprises (Foster, Kesselman, Nick, Tuecke, 2002). To date, many distributed computing paradigms exist, such as Common Object Request Broker Architecture (CORBA), Java's' Remote Method Invocation (RMI), Component Object Model (COM), Web services, etc. Grid Services are an evolution of existing paradigms (Foster, C. Kesselman, S. Tuecke, 2001). The use of open standards such as Open Grid Service Infrastructure (OGSI), extensible Mark-up Language (XML) and Simple Object Access Protocol (SOAP) easily allow for heterogeneous platforms to communicate and share computing resources within a virtual organisation (VO) context.

Grids are a relatively new concept. The term Grid, in popular perception, has been loosely used to describe a range of concepts, anything from advanced networking to Artificial Intelligence. Grids are primarily concerned with "coordinating resource sharing and problem solving in dynamic, multi-institutional virtual organizations" (Foster, et al, 2001). There is often a need to integrate service across distributed, heterogeneous, dynamic "virtual organizations". This sort of integration can be technically challenging due to the need to achieve a certain level of Quality-of-Service (QoS) on top of different native platforms (Foster, et al, 2002).

The Open Grid Service Architecture (OGSA) specification identifies Grid requirements and capabilities for building Grids. The definition of the OGSA specification is very closely associated with Web-service standards, such as Web-Service Resource framework (WS-RF). Grids attempt to leverage Web service technologies to provide platform independent interoperable services, capitalizing on desired Web-service properties (Foster, et al, 2002). These desired Web service properties include: service description and discovery, automatic generation of client

and server code from service descriptions, binding of service descriptions to interoperable network protocols, compatibility with higher level open standards, services and tools, and broad commercial support (Ibid).

OGSA Grids aim to provide open interoperable services to facilitate the creation and management of scalable virtual organizations. This poses some strong security challenges to Grid designers. Grids can often be made up of participants from multiple physical organizations. Grid participants might make use of incompatible (heterogeneous) underlying platforms and technologies, and security policies.

The goal of this research is towards a holistic Grid security framework. This research will propose two possible abstract layers of a Grid, for this purpose, and focuses primarily on OGSA-based Grid systems. These layers will be defined along with their corresponding security challenges.

1.1 Motivation For This Study

1.1.1 Realization That Grid Technology Is A Rapidly Growing Technology

Grids encompass evolving state-of-the-art technologies that will continue to have a large impact on the computer industry. MIT technology review (2003) has identified OGSA Grids as a technology that will change the world in the way that we do business and live our lives. Industry leaders, such as Sun Microsystems, Hewlett-Packard (HP), Microsoft, and Oracle, are adopting Grid technologies and plan on including Grid capabilities into their products. Oracle has already started building Grid capabilities into their commercial products (Kontzer, Whiting, 2004). Grid technologies have been included in their 10g family of products (Kusnetzky, Olofson, 2004).

1.1.2 Realization That There Is A Need For Interoperable, Standards-based Grid Security Solutions

With Grid specifications constantly under modification and review, Grid security in particular is in need of attention. Grids are primarily a technology to facilitate scientific and commercial collaborations between various parties, forming virtual organizations (VOs). These collaborations are often over large physical distances and participants typically utilize heterogeneous platforms. Grid middleware provides a layer of interoperability on top of existing infrastructure, to support the integration and management of resources within VOs. These interactions across disparate trust domains present a number of security challenges.

1.2 Problem Statement

Grid security continues to be a new area of research. There are currently few adequate Grid security solutions that address issues on all levels of (variable) Grid architecture. To compound the problem, Grid specifications are under constant review and modification. Standardised Grid security practices and specifications are still lacking. This is largely since current solutions are adapted to Grids which evolve from pre-existing infrastructure, rather then being specifically designed for "generic" Grid requirements. Grid Security Infrastructure (GSI) is a "generic" Globus specification which will be re-visited in subsequent chapters, but encompasses inherent problems. Grid security challenges are divided into two distinct categories: political (inter-organisational) issues, and technology (Grid fabric) issues.

Higher level (political) issues stem from Grids operating across organizational and administrative domains. Grids need to provide coherent and adequate authentication, authorization, and to facilitate complex trust relationships across these various domains. This is a challenging task, considering participants abide by conflicting, or incompatible policies.

Lower level Grid security is concerned with securing Grid resources. These are the resources that collectively make up a Virtual Organisation (VO). Grids do not prescribe standard underlying hardware or software. Although Grid resource

threats can be identified, no standard solution can be recommended that will be relevant to all Grid implementations (due to the diverse nature of Grid resources). Standard guidelines are needed to address these security concerns. These guidelines need to have relevance to all Grid implementations.

This research will attempt to define a holistic framework to secure OGSA-based Grids. Both political and lower level Grid security concerns will be discussed and addressed. To successfully define such a framework, the following research questions need to be answered initially (mainly, but not solely, from a security perspective):

- What is a Grid?
- How do Grids differ from other distributed computing paradigms?
- What are the Grid requirements and capabilities defined in the OGSA specification?
- How do current versions of the Globus toolkit factor in OGSA requirements and capabilities?
- What technologies enable interoperable Grid messaging?

1.3 Objectives Of This Study

The primary objective of this research is to propose a framework towards holistic Grid security. The framework is intended to address both higher level political issues and lower level technological issues. To achieve this objective, the following sub-objectives have, among others, been pre-empted:

- A generic method for identifying Grid resource threats must be defined. This method must be applicable to a wide range of hardware and software configurations. Relevant countermeasures to Grid threats must be identified.
- A single political-level authority to facilitate authentication, authorization, and trust relationships between Grid participants must be defined.

1.4 Methodology

The primary methodology that will be utilized during this study is scientific argument (roughly based on the phenomenological approach), underpinned by a comprehensive literature survey. The literature survey will examine the current Grid landscape by examining relevant work published in the domain of discourse. The nature of Grids, as espoused in Grid specifications and supporting standards, will be examined. This, together with previous research will be used to find critical success factors which define a holistic Grid security framework.

Early reading in the field indicates that one can distinguish two layers of security challenges that exist in the Grid environment. An approach to address the political Grid security challenges, which allows for diverse fabric-level implementations, will be identified. Also, a generic threat model will be defined. The literature survey will include a discussion on how this model must be generic enough to be applicable to the heterogeneous nature of Grid resource implementations, but must take Grid specific constraints into account.

The results of this study have been reported both in this dissertation and an academic paper.

1.5 Overview Of The Dissertation

The proposed layout of the dissertation is depicted in figure 1.1. The dissertation consists of eight chapters.

Chapter 1 provides introductory information on the problem area of the dissertation.

Chapter 2 aims to introduce the salient concepts of Grids. The chapter compares common distributed commuting technologies to Grids, and highlights the differences between them.

Chapter 3 discusses the Open Grid Service Architecture (OGSA) and the Globus toolkit. It discusses the OGSA 1.5 specification in detail, highlighting relevant sections of the specification. The Globus toolkit is also discussed in some

detail. The Globus toolkit is based on OGSA. This chapter will discuss how Globus implements the OGSA specification

Chapter 4 discusses concepts and technologies that support OGSA-based Grids. Grids are based on a Web-service Service Orientated Architecture (SOA). SOA is highlighted and discussed with relevance to Grids. Supporting Web-service technologies such as Web-service Resource framework (WS-RF), Web-service Description Language (WSDL), Simple Object Access Protocol (SOAP) are also discussed.

Chapter 5 discusses the need for Grid security. STRIDE, as a threat classification scheme, is introduced and discussed. This chapter proposes two logical layers of a Grid, based on the Grid security requirements. Security concerns in these layers are addressed in chapter 6 and 7.

Chapter 6 defines a broker entity to facilitate high level Grid security challenges. The broker is defined is an abstract high level software component. The broker addresses the 'political' security issued identified.

Chapter 7 discusses a generic threat-modeling methodology in terms of the generic steps taken by an attacker (from a Grids perspective). This chapter outlines a generic threat-modelling process that can be applied to almost all Grid resource implementations.

Chapter 8 concludes the dissertation.

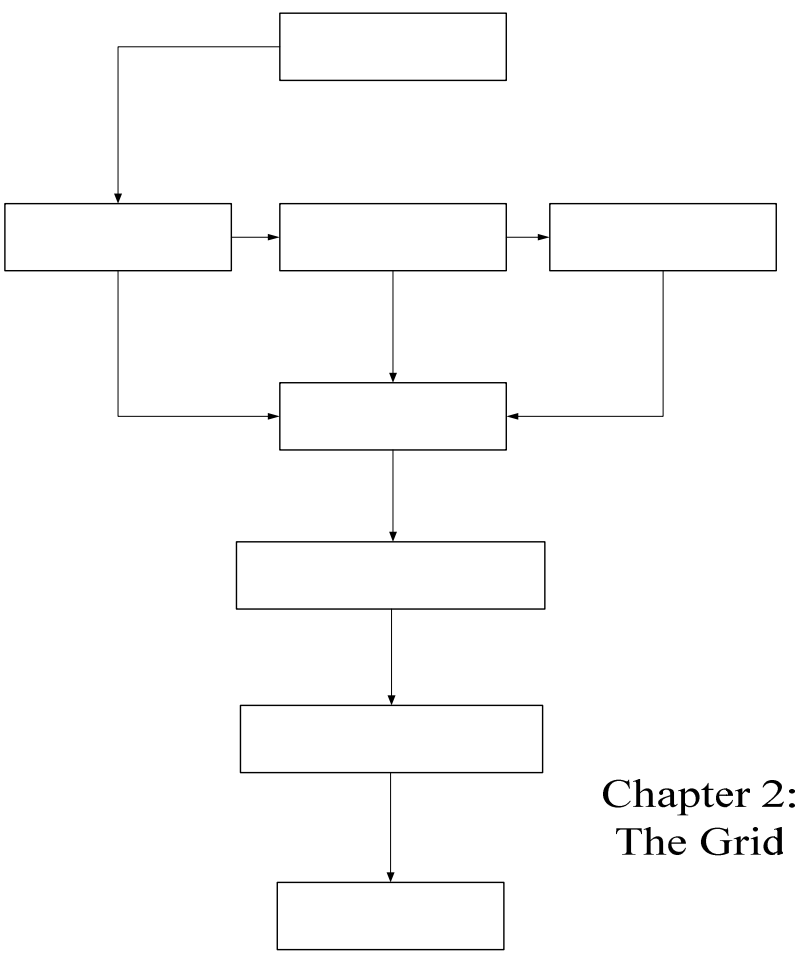

Figure 1.1: Proposed layout of the dissertation

Chapter 2

The Grid

In recent years the concept of Grid computing has gained in popularity. Grids allow for large scale collaboration between dispersed parities, which typically make use of heterogeneous platforms and technologies. The term "Grid" has been largely misrepresented. To fully understand the capabilities and benefits of utilizing Grid technologies, one must have a clear understanding of what Grids are, what Grids clearly are not, and what the goal of Grid computing is.

The following is an early definition of Grid computing, "A computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." (Foster, Kesselman, 1998; Berman, Fox, Hey, 2003). Recent research in the Grid area has identified several newer layers, on top of the hardware and software layers identified in earlier Grid definitions. A Grid might not be limited to a single physical organization and administrative body; thus, new challenges are introduced. Grids can be applied to inter-organizational collaborations, know as virtual organizations. There are social and policy issues to be considered within cross-organizational Grid implementations, above the technical layers required to make Grids work.

The goal of Grid technologies is, "to coordinate resource sharing and problem solving in dynamic, multi-institutional virtual organizations" (Foster, et al, 2001). Although the primary goal of Grids is to promote multi-institutional collaborations in a virtual organization, this is not the exclusive application of Grid computing. Grid computing concepts can be applied within single organizations as well. Organizations might want to couple arrays of network nodes together to provide powerful computing structures, or create powerful knowledge-bases utilizing Grid enabled technologies. The key Grid concept, is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers), and then to use the resulting resource pool for some common purpose

(Foster, 2002). Grids need to be able to facilitate these requirements on top of heterogeneous hardware and software platforms (provide interoperability). According to Foster, a Grid should be evaluated in terms of its applications, business value, and scientific results it delivers, not its architecture (Ibid).

There are common misconceptions around what constitutes a Grid. Peer-to-peer (P2P) networks and clustering, for example are technologies that have similar goals, but are different in focus of design, requirements and communities (Foster, Iamnitchi, 2003).

The next three sub-sections are intended to clarify the Grid concept.

2.1 What Is The Grid?

Goble makes the analogy of a computational Grid to a power Grid (electrical Grid), "computing and data resources would be delivered over the Internet seamlessly, transparently and dynamically as and when needed, just like electricity" (Goble, De Roure, 2002). The Grid technologies aim to provide seamless and consistent resource publishing, discovery, and access, across heterogeneous hardware and software environments.

A Grid provides resource sharing and collaboration capabilities to dispersed parties participating in a virtual organization context. In order for Grids to function as they are intended to, Grids require protocols (and interfaces and policies) that are not only open and general-purpose but also standard (Foster, 2002). Foster defines a three point checklist as to what a Grid is:

1. Coordinates resource sharing that is not subject to centralized control
2. Using standard, open, general purpose protocols and interfaces
3. To deliver non-trivial Quality-of-Service

According to Foster, "the Grid is not a monolithic client-server structure..." and "…a primary characteristic of Grids is to not be subject to central control" (Ibid). However, some form of central control is necessary to achieve coordinated resource sharing. A Grid can be deployed over several sites worldwide, making use of incompatible underpinning technologies, such as platform and security

technologies. It will be argued in this dissertation that there is a need to provide some method of "central" control in the form of a broker (discussed in chapter 6), although its control is not absolute. A broker in a Grid context acts as a mediator between Grid participants, and facilitates communication through open standards and middleware. Standardization is one of the key aspects to Grid computing:

- "It is standards that allow one to establish resource-sharing arrangements dynamically with any interested party and thus to create something more than a plethora of balkanized, incompatible, non-interoperable distributed systems" (Ibid).

Efforts have been made by the Global Grid Forum (GGF) and other interested parties to provide standards for Grid implementations. The Open Grid Service Architecture (OGSA) is a widely accepted specification for defining the Grid capabilities required by Grid middleware. Grid middleware is a term to describe the tools and APIs necessary to facilitate Grids, i.e. the software layer needed to provide interoperability among heterogeneous platforms. OGSA is a constantly evolving specification. The Globus toolkit, currently in its fourth revision (GT4), is an open-source toolkit, based on the OGSA specification. The GT4 contains tools and services for implementing Grids and Grid resources.

OGSA prescribes the WS-* set of Web-service standards, for publishing services and service discovery. The Open Grid Service Architecture (OGSA) builds on Grid and Web-service concepts, to provide a set of standard capabilities for publishing Grid resources. OGSA defines standard mechanisms for creating, naming, and discovering transient Grid resource instances; furthermore, it provides location transparency and multiple protocol bindings for service instances; and supports integration with underlying native platform facilities (Foster, Kesselman, Nick, Tuecke, 2002b).

2.2 The Goal Of Grid Computing

The Grid computing concept is an evolution of traditional distributed computing paradigms. Grids have a focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation. Foster identifies the goal of Grid computing as, "the real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations" (Foster, et al, 2001). Current Internet technologies primarily address communication and uniform information exchange between computers, but do not adequately facilitate the coordinated use of resources available at multiple co-operating sites. Grids facilitate the creation of dynamic sharing relationships between any potential participants. Grid participants do not make use of a prescribed set of hardware and software; so, many Grid participants utilize heterogeneous platforms. Thus, interoperability is a central issue that Grids need to address.

Grid concepts and technologies were initially developed to establish resource sharing within scientific collaborations (Foster, et al, 2002). In recent years there has been a growing trend towards commercial applications for Grids as well, just as the Internet World Wide Web (WWW) began as a technology for scientific collaborations and was later adopted for commercial mainstream use. The same is expected for Grid applications (Ibid). Grid technologies are also concerned with providing resource sharing, and harnessing existing resources and infrastructure to satisfy new, emerging business needs within single organisations. With the emergence of the Internet as a business tool, there has been a growing realization that companies' IT infrastructure also encompasses external networks, resources, and services (Ibid).

Grid systems aim to integrate, virtualize, and manage services and resources, within distributed, heterogeneous virtual organizations (Foster, Savva, Berry, Djaoui, Grimshaw, Maciel, Siebenlist, Subramaniam, Treadwell, Von Reich, 2006). The Grid paradigm faces many challenges including: authentication, authorization, resource access, resource discovery, etc, within a virtual organization. Individual (or physical) organizations that are included or have

membership to a collaborating virtual organization often have diverse security infrastructure and security policies. There is a need for higher-level (abstracted) services to provide interoperability among diverse participants.

Grids provide this interoperability by utilizing middleware. Middleware is defined as, "the services needed to support a common set of applications in a distributed network environment" (Aiken, Strassner, Carpenter, Foster, Lynch, Mambretti, Moore, Teitelbaum, 2000). Grid middleware is intended to be easily implemented and complementary to existing network infrastructure and services within the adopted organization, or organizations participating in a virtual organization context. Grid middleware enables proxies for standardized communication channels between participants in a Grid, but this occurs on a local (fabric) level. Grid political-level requirements include that resources be: discovered, accessed, allocated, accounted for, etc. In general, all these entities need to be managed as a single virtual system. This should be possible, even when the hardware\software infrastructure is provided by different vendors and/or managed by different organizations. Standardization is crucial for the creation of interoperable, portable, and reusable Grids. The same applies to Grid security considerations; a political-level brokered architecture will be argued for in this study.

To summarize, the primary goal of Grid systems is to facilitate resource sharing in a large collaboration of diverse parties. The standardization of protocols and services is required to support secure Grid: authentication, authorization, service discovery, and service publishing, in diverse heterogeneous environments. Just as the World Wide Web (WWW) and other earlier network technologies were originally utilized purely for scientific collaborations, and later adopted for commercial gain, the same is expected to occur with Grid technologies.

2.3 How The Grid Differs From Similar Existing Paradigms

The previous section described what a Grid is, and what the goals of Grid technology are. This section will compare other prominent modern resource\information sharing technologies to the Grid. Several technologies will be identified. Their differences and similarities to Grids will be discussed.

The following elements of a Grid have been identified, and were discovered to be crucial and in a combination unique to Grid systems:

- Coordinated resource sharing
- Not subject to centralized control
- Utilizing open, general purpose protocols
- Delivers a non-trivial Quality of Service
- Interoperable among diverse platforms and technologies
- Provides adequate security services

The above mentioned criteria for a Grid will be compared to the following resource\information sharing technologies. This will establish that existing technologies in current widespread adoption cannot fully satisfy all the Grid requirements (Foster, et al, 2001; Foster, et al, 2003).

1) World Wide Web (WWW)
2) Third-party service providers
3) Enterprise computing systems
4) Internet and peer-to-peer (P2P)

2.3.1 World Wide Web (WWW)

The World Wide Web (WWW) is a powerful tool for sharing information. The WWW is built on rich technologies for sharing information. The efforts made by the IETF and W3C have seen the rise of standards and protocols, which make the WWW an attractive platform for constructing virtual organization systems and applications. These technologies include HTTP, HTML, TCP\IP, and XML.

These technologies are excellent at supporting client-browser to Web-server interactions. However, on their own, they lack the richer interaction capabilities required for modern virtual organizations. For instance, Web content is generally subject to centralized control, following a client-server interaction model.

2.3.2 Third-party Service Providers

Third-party service providers typically provide outsourced IT services, such as specialized business applications and storage capabilities, among others. Interactions between third-party providers and clients are often an on-request, client-server interaction. There is generally a pre-established service level agreement (SLA) between the provider and client, defining the access to hardware and service combinations.

From the perspective of a virtual organization, third-party service providers provide building blocks for a VO. Virtual Private Networks (VPNs) and static configurations are inherent to the type of relationship that exits with a third-party service provider. This makes a coherent VO resource sharing model hard to achieve.

The static nature of third-party utilities makes it difficult, almost impossible to create smart, Grid-enabled applications; for example, if there is a dataset stored on a storage service provider (SSP) site, and an application hosted at an application service provider (ASP) site. The application hosted at the ASP will not dynamically learn of the dataset, its content, or the security requirements to access it.

The integration of Grid technologies into third-party services, provided by storage service providers (SSP) and application service providers (ASP), could provide a richer range of services to organizations. Third-party services alone are merely building blocks for a virtual organization.

2.3.3 Enterprise Computing Systems

Enterprise development technologies such as Common Object Request Broker (CORBA), Enterprise Java Beans, Java 2 Enterprise Edition (J2EE), and Distributed Component Object Model (DCOM), are all paradigms available to create distributed enterprise applications. These paradigms provide: resource interfaces, remote method invocation mechanisms, service publishing and service discovery. These services make it easy to share resources within a single organization but require pre-agreement for inter-organisational interactions.

Enterprise computing systems provide resource sharing; however, they do not satisfy Grid requirements. Enterprise computing systems are relatively static and are restricted to occur within a single organization. Their primary form of interaction is typically client-server, rather then the coordinated use of multiple resources.

2.3.4 Internet And Peer-to-Peer (P2P)

Grids and Peer-to-Peer (P2P) are both concerned with the pooling and coordinated use of resources within distributed communities (Foster, et al, 2003). Grids and P2P technologies share similar end goals. However, there are some fundamental differences in their implementations and application.

Grids are concerned with providing a rich set of resources, to restricted communities, delivering non-trivial Quality-of-Service to its users. Peer-to-Peer, however, provides a small set of services, to a wider user base, with out any real concern for Quality-of-Service (QoS), delivery and trust. The following table illustrates some of the differences between Grids and P2P:

Table 2.1: P2P and Grid comparison

	Grids	**P2P**
Community base	Smaller	Larger
QoS concerns	Yes	No
Services provided	Rich set	Basic
Trust between participants	High amount required	No Concern

2.4 Conclusion

This chapter addressed the fundamentals of Grid technologies. It sought to demonstrate the goal of Grid computing and how the Grid paradigm differs from other existing technologies for sharing resources and providing interaction beyond the single enterprise.

Chapter 3

OGSA and The Globus Toolkit

Chapter 2 introduced the concept of Grids for implementing resource sharing, and distributed systems across organizational boundaries, thereby forming virtual organizations (VOs). Open Grid Service Architecture (OGSA) is a specification aimed at standardizing the Grid paradigm (Foster, et al, 2006). The OGSA specification (version 1.5) will be used as a basis for the discussion in this chapter. The Globus Toolkit (GT4; version 4), an open community project implementing many of the requirements and capabilities defined by OGSA, will be discussed as well.

The Open Grid Service Architecture (OGSA) is a Service Orientated Architecture (SOA). SOA will be discussed in more detail in Chapter 4. In chapter 2, it was highlighted that standardization is the key to allowing heterogeneous systems to be discovered, accessed, monitored, and managed as a single, virtual system. OGSA builds on technologies from both the Grid and Web-service communities. Technologies form the Web-service community allow Grid designers to make use of standardized, platform independent interfaces for building Grids on a variety of native operating system platforms. The definition of the OGSA specification is closely tied to the WS-* set of specifications. WS-* are a set of specifications for implementing service orientated Web-services. The particular WS-* specification of interest to this research is the WS-Resource Framework (WS-RF), which will be discussed in chapter 4.

The OGSA specification is divided into two main sections, OGSA requirements and OGSA capabilities. The specification provides a set of abstract requirements that OGSA is intended to address. These requirements are translated into a set of capabilities that collectively define OGSA (Foster, et al, 2006).

The capabilities defined by OGSA are implemented in an open source project called Globus. The Globus Toolkit is a set of services and components for implementing OGSA compatible Grids. The Toolkit is currently in its fourth

version. Globus is the de facto standard for Grid implementations (Gerndt, 2004). Security is implemented in the Globus Security Infrastructure (GSI); this component of the Globus Toolkit is of particular interest to this research as it is one of the few practical implementations of a Grid security infrastructure. GSI makes use of X.509 certificates to authenticate and authorize parties within a Grid context (X.509 is discussed in Chapter 7).

The following sections will provide detailed discussions of the OGSA requirements and capabilities, and a brief discussion on how Globus implements OGSA capabilities. WS-RF will be briefly discussed in Chapter 4.

3.1 OGSA Requirements

OGSA requirements are driven by a set of functional and non-functional requirements. These requirements are largely based on use cases identified in the OGSA specification (Foster, et al, 2006). The following table provides a list of the OGSA requirements and a brief summary of each requirement (Ibid):

Table 3.1: A brief summary of OGSA requirements

OGSA requirement	Summary
1. Interoperability and support for dynamic heterogeneous environments	OGSA must provide interoperability between such diverse, heterogeneous, distributed resources and services, as well as reducing the complexity of administering heterogeneous systems.
2. Resource sharing across organizations	One of the main goals of OGSA is to enable resource sharing and virtualization across administrative domains
3. Optimization	Optimization refers to the technique used to allocated resources effectively to meet consumer (client side) and supplier (server side) needs.
4. Quality-of-Service assurance	Services and Grid resources must provide clients with the agreed upon Quality-of-Service
5. Job execution	OSGA must provide manageability for execution of user defined tasks throughout out their lifetime
6. Data services	OGSA must provide efficient access to large datasets, as well as the abilities to move them between Grid participants
7. Security services	Safe administration of distributed resources,

	requires controlling access to resources through secure and robust security protocols
8. Administrative cost reduction	Consistent and automated management operations are required, in order to minimize cost and the possibility of human error
9. Scalability	The large scale nature of Grid systems might put novel demands on the management infrastructure, the management architecture needs to be able to scale with the growth of the Grid
10. Availability	Disaster recovery mechanisms are needed to ensure the operation of a Grid system can be recovered quickly and efficiently in case of natural or human-caused disaster
11. Ease of use and extensibility	OGSA enabled Grids should mask the complexity of the environment from its users

The following sub sections will discuss these requirements in more detail.

3.1.1 Interoperability and Support For Heterogeneous Environments

Grid environments are often large and dynamic, encompassing heterogeneous and largely distributed parties. Grid participants make use of a variety of hosting platforms (.NET, JAVA, etc), operating systems (Windows, LINUX, UNIX, etc), devices, and services. Grids are intended to be long-lived and dynamic, and could possibly evolve past the original design specification.

OGSA is required to cater for these scenarios. It must provide interoperability between these dynamic, heterogeneous, and largely dispersed parties; while reducing the complexity of managing such environments.

The following requirements to support heterogeneous systems are defined in the OSGA version 1.5 specification (Foster, et al, 2006):

- **Resource virtualization –** Resource virtualization is essential to reduce the complexity of managing heterogeneous and diverse systems, and to handle diverse resources in a unified way.
- **Common management capabilities –** Common management methods are required to simplify management of heterogeneous systems. Uniform and consistent management methods are required.

- **Resource discovery and query –** Mechanisms within OGSA are required for identifying resources with capabilities required by Grid users.
- **Standard protocols and schemes -** Standard protocols are requirements for interoperability.

3.1.2 Resource Sharing Across Organizations

Grids are not a monolithic system, but often consist of resources owned by multiple organizations (Foster, et al, 2006). OGSA must be able to support resource sharing across administrative domains, even across organizational boundaries. Cross-organizational resource sharing requirements include (Ibid):

- **Global namespace –** Global namespaces allow for simplified data and resource access. Global namespaces provide unique identification for Grid participants.
- **Metadata service –** Metadata services are required for finding, invoking and tracking entities. Metadata services are required to provide information about Grid entities and their current state.
- **Site autonomy -** Resources must be accessible across sites. However, local control and policy must still be respected.
- **Resource usage data -** Standard mechanisms for collecting and distributing resource usage information across organizations, for the purpose of accounting, billing, etc is required.

3.1.3 Optimization

Optimization as defined in the OGSA specification is as follows, "Optimization refers to techniques used to allocate resources effectively to meet consumer and supplier requirements" (Foster, et al, 2006). OGSA must make optimization considerations for both consumers and suppliers participating in a Grid. The OGSA specification refers to consumers as a 'client' or service requestor, while a supplier is referred to as a service provider. An example of optimization would be client-side caching of data to improve network performance.

3.1.4 Quality-of-Service Assurance

Grid resources must provide an agreed-upon Quality-of-Service (QoS) between Grid consumers and suppliers. OGSA defines key QoS dimensions, such as availability, security and performance. QoS expectations must be expressed using measurable and commonly understood terms. OGSA QoS assurance requirements include (Foster, et al, 2006):

- **Service level agreements -** QoS should be represented as an agreement between provider and requester, prior to service execution. Standard mechanisms should be provides to create and manage QoS agreements.
- **Service level attainment –** Mechanisms must be provided to ensure attainment of agreed upon service level agreements between Grid resource consumers and suppliers.
- **Migration -** It should be possible to migrate executing services or applications to adjust workloads for performance or availability.

3.1.5 Job Execution

OGSA must provide flexibility and manageability for executing user defined jobs (processes), thought out the lifetime of the job. Furthermore, functions such as scheduling, provisioning, job control and exception handing of jobs must be supported thought out the processes lifetime; even if the process is distributed across heterogeneous resources. OGSA defined job execution requirements (Foster, et al, 2006):

- **Support for various types of jobs -** Executions of various types of jobs must be supported including simple and complex jobs, such as workflow and composite services.
- **Job management -** It is important to be able to manage jobs during their entire lifetime. Jobs must support manageability interfaces and must work with various types of groupings of jobs.
- **Scheduling -** The ability to schedule and execute jobs based on priority, and current resource allocation (capacity) is required.

- **Resource provisioning -** Measures need to be put in place to automate the process of resource allocation, deployment and configuration.

3.1.6 Data services

Data services must provide efficient access to distributed datasets, and the ability to move and manage them. OGSA must simplify the creation of data-orientated applications, and make them resilient to changes in the Grid environment (Foster, et al, 2006). OGSA defines the following data service requirements (Ibid):

- **Policy and specification management -** Policies are required to define how the data is accessed and managed in a Grid environment.
- **Data storage -** Storage for Grid data is required; the most typical form of storage are hard disk drives. Common interfaces provide common storage and management.
- **Data access -** Easy and effective access to various types of data (database, file, and streams), independent of its physical location or platform, by abstracting underlying data sources, is required.
- **Data transfer -** High bandwidth data transfer is required. This requirement is an infrastructure requirement. High speed networks and redundant network paths are required. Redundant network paths provide load balancing when the infrastructure is under strain.
- **Data location management -** These services manage where data is physically stored.
- **Data update -** Grids must provide updated facilities that maintain consistency of updated datasets. These services must ensure the data is correct, consistent and up-to-date.
- **Data persistency –** All data and metadata should be maintained for the entire lifetime of the Grid user request.
- **Data federation -** Federation of data across heterogeneous environments should be supported. Heterogeneous data might be organized in different schemes, or stored using different technologies. Mechanisms to convert and federate data interactions between heterogeneous platforms are required.

3.1.7 Security Services

Standard and secure mechanisms are required to secure Grid interactions. Grids need to support safe resource-sharing across different administration domains. OGSA defines standard security requirements (Foster, et al, 2006):

- **Authentication and Authorization -** Authentication is required to identify individuals and services within the Grid. Consistent authorization assertions are required to be consistent throughout all layers of the Grid.
- **Multiple security infrastructures -** Distributed operations imply the need to integrate multiple security infrastructures. OGSA must be able to integrate and be interoperable with existing security architectures and models.
- **Perimeter security solutions -** Resources may be accessed across organizational boundaries. OGSA requires standard and secure mechanisms that can protect organizations, and yet allow for secure cross organizational collaboration.
- **Delegation –** User rights must be delegate-able to user processes. A process should be able to utilize resources on behalf of the user executing the process.
- **Security policy exchange -** Service requestors and providers should be able to dynamically share policy information, to allow the establishment of a negotiated security context between them.
- **Intrusion detection, protection, and secure logging -** Strong monitoring of intrusions and misuse is required in order to help mitigate security incidents.

3.1.8 Administrative Cost Reduction

There are high financial costs, and an increased possibility of human error when administering large scale complex distributed environments. OGSA automates standard administrative tasks for Grid administrators. OGSA defines the three following methods for reducing the administrative costs associated with Grids (Foster, et al, 2006):

- **Policy based management** – Grid administration could be automated at all layers of the Grid. This includes low level technology policies, to higher level process policies.
- **Application contents management -** Application contents management can allow for the deployment, configuration and maintenance of complex systems. This approach will allow for concise and reliable management of components, without expert knowledge of the applications.
- **Problem determination (troubleshooting) -** Troubleshooting mechanisms are required, so administrators can quickly recognize, cope with and fix emergencies.

3.1.9 Scalability

The large scale distributed nature of a Grid could put strain on the management infrastructure. The management architecture needs to scale to potentially support thousands of heterogeneous resources.

3.1.10 Availability

A high level of availability is a requirement in high performance Grid environments. A high level of availability can be achieved through fault tolerant hardware. In the case of data loss or services loss, disaster recovery mechanisms can be employed to ensure speedy service continuation.

3.1.11 Ease Of Use and Extensibility

OGSA should mask the complexity of the environment from its users (Foster, et al, 2006). The Grid must provide extensibility and customization in a way that does not compromise interoperability.

3.2 OGSA Capabilities

The following section will discuss the OGSA capabilities as they are represented in the OGSA specification version 1.5 (Foster, et al, 2006). Version 1.5 of the specification was the latest version at the time this document was authored. OGSA capabilities define a set of services to address the requirements outlined in the document. These requirements were discussed in the previous section.

OGSA defines a set of capabilities that allow for the seamless use and management of distributed heterogeneous resources. OGSA defines three logical tiers to a Grid, these tiers are as follows:

- base resources (bottom tier)
- virtualization and abstraction (middle tier)
- applications (top tier)

Base resources are supporting underlying resources. These resources could be logical or physical resources, which have relevance outside OGSA. These resources include hardware (CPU, memory, disk space), or OS processes, etc. and are often referred to as the *Grid fabric*. According to Foster, "This layer (resource tier) provides the resources to which shared access is mediated by Grid protocols" (Foster, et al, 2006).

The *virtualization and abstraction* tier is made up of Grid middleware that facilitates the interactions between Grid participants. This tier implements OGSA capabilities and services. These capabilities support applications and processes, on the highest level of the Grid architecture. A detailed relationship between the middle tier and lower tier (or base resource level) exists.

The applications layer is a logical representation of applications and processes. This tier builds on the two lower tiers to realize user and domain orientated processes and functionality (such as business processes).

OGSA services and capabilities are mostly realized in the 'virtualization and abstraction' tier, or middle tier. The following sections will discuss the services and capabilities required by OGSA to facilitate the creation, use and management of Grid resources in a virtual organization context.

3.2.1 Infrastructure Services

OGSA shares and builds on a number of common services. Current work on OGSA builds on, and contributes to, the growing set of Web-service architecture standards. The WS-* set of standards provide OGSA Grids with a robust service orientated architecture, in particular the Web-service resource framework (WS-RF). According to the OGSA specification, "Web-services Architecture is the most effective route to follow to achieve a broadly adopted, industry-standard service-oriented rendering of the functionality required for Grid systems" (Foster, et al, 2006).

Web-service standards utilized in conjunction with OGSA, allow OGSA to provide a service-orientated architecture. Web Service Description Language (WSDL) is used to define service interfaces (Christensen, 2001), and Simple Object Access Protocol (SOAP) is utilized as the primary message exchange format between OGSA resources. SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment (Gudgin, Hadley, Mendelsohn, Moreau, Nielsen, 2003). Both SOAP and WSDL are discussed in Chapter 4.

The combination of these technologies provides a foundation for building complex Grid resources. Grid resources are consumable by a variety of software and hardware platforms. These technologies promote interoperability in large, widely dispersed, heterogeneous Grids.

3.2.2 Execution Management Services (EMS)

Execution Management Services (EMS) is primarily concerned with executing and managing units of work, until the completion of the job. OGSA specification defines units of work as follows, "units of work may include either OGSA applications or legacy (non-OGSA) applications (a database server, a servlet running in a Java application server container, etc)" (Foster, et al, 2006). OGSA defines the following objectives of Execution Management Services (Ibid):

- EMS must, find execution candidate locations,
- select execution candidate locations,
- prepare execution,
- initiate execution, and
- manage executions for the duration of their lifetime.

3.2.3 Data Services

OGSA data services provide the capabilities necessary to move and manage data as required in a Grid environment. Data services are accessible by other OGSA defined services that require access to data. All direct interactions with data are handled by the OGSA data services. Data services are able to interpret and store data in different formats. Due to the heterogeneous nature of Grids there are no standard methods of storing data. Some examples of possible data resources available in Grids include: flat files, streams, a variety of Database Management Systems (DBMSs), data catalogues or directories.

3.2.4 Resource Management Services

Grid Resource Management Services manage Grid resources. There are three types of resource management in Grids. OGSA defines them as follows (Foster, et al, 2006):

- Management of the underlying resources (Grid fabric),
- Management of OGSA Grid resources, and
- Management of the OGSA infrastructure (Grid middleware).

3.2.5 Security Services

Grids potentially cross administrative domains. Foster defines broad Grid security requirements as follows:

- "OGSA security architectures must support, integrate, and unify popular security models, mechanisms, protocols, platforms, and technologies in a way that enables a variety of systems to interoperate securely" (Foster, et al, 2006).

OGSA security models must be able to plug into and be compatible, with a wide range of security architectures and models across a wide range of hosting platforms and operating systems. Single organizational domains within a virtual organization tend to implement their own local security policies to achieve their individual business goals. These security policies could vary in implementation and strictness. All interactions between parties in a Grid are subject to Grid security policies, and the local security policies of interacting participants.

Grids security services facilitate the enforcement of policy-based security architecture. The enforcement of policy-based security is to ensure the higher level business objectives are met. Grids must provide the following security services: authentication, identity mapping, authorization, credential conversation, audit and secure logging, and privacy (Foster, et al, 2006). Many of these security requirements are addressed in the framework proposed by this research.

3.2.6 Self Management Services

Self management is a concept that reduces the cost and complexity of maintaining large IT infrastructure. Self managing environments allow for components, hardware and software, to troubleshoot themselves. These components can identify faults and correct their configuration; or notify administrators of a problem, allowing them to solve it problem proactively. OGSA defines a set of self management service, but also indicates that not all participants and resources will make use of all or any of the defined services.

3.2.7 Information Services

Information services stores metadata about Grid resources. These services allow Grid users and services to access and manipulate information about Grid resources. Information services provide a directory of static Grid resource information, and current and dynamic information. Metadata about a service, such as capabilities, and security requirements, are accessible via OGSA information services.

3.3 Globus Toolkit

The Globus Toolkit is an open source implementation of all the protocols and primitives defined by Open Grid Service Architecture (OSGA), for implementing Grid resources (Sandholm, Gawer, 2003). The Globus Toolkit is currently in its fourth revision (GT4). The toolkit consists of a number of components based on Grid requirements and capabilities defined by the OGSA specification. The OGSA specification was discussed in the previous sections. This section briefly discusses OGSA requirements within a Globus context.

Globus makes use of a layered architecture; high level global services are built on a core set of lower level services (Foster, Kesselman, 1998b). One of the most important services within the Globus Toolkit is the resource management service. *Globus Recourse Allocation Manager (GRAM)* is responsible for allocating and de-allocating resources to services.

In most distributed system architectures, communication plays a key role; the Globus Toolkit provides a communication component, called *NEXUS*. NEXUS is a library of lower level communication APIs that provide support for higher level communication (Ibid).

Security is a major concern for Grid implementations. Grid security requirements are diverse. *Globus Security Infrastructure (GSI)* is the component within the toolkit that provides security. GSI mostly addresses the problem of authentication, and therefore leaves open a large area for future research in the space of Grid security (Ibid).

In dynamic environments such as Grid systems, there is a need to easily access information about services, components, and applications, in a timely manner. Having this information available, allows the Grid to adapt to changes in system structure and state. *Globus Meta-Computing Directory Service (MDS)* stores and makes the following information accessible to Grid participants: architecture information, operating system information, memory available on a network node, network bandwidth and latency, communication protocols, the mapping between IP addresses and network technology (Ibid). MDS provides tools and APIs to allow for discovery, publishing and access to information about the structure and state of a Grid resource.

Health Beat Monitor (HBM) provides simple management services for monitoring the health and status of remote processes. The HBM consists of several client APIs. Grid processes register with the HBM upon initialization. HBM then acts as a data-collection base, it periodically receiving "heart-beat" information about a process. Other processes can query the HBM of the status process.

Globus provides *Global Access to Secondary Storage (GASS)* this component provides Grid applications with access to simple C I\O libraries; and the ability to open, edit, save files on remote computers.

Globus Executable Management (GEM) supports remote identification, creation and location of executables in heterogeneous environments. This service is limited in the current version of the Globus Toolkit.

This section has highlighted several main components within the Globus Toolkit, each with its own purpose and name within the toolkit. The following is a summary-list of these components, and their commonly used abbreviated names:

- **Resource management (GRAM)** – Allocates resources to jobs and performs process management.
- **Communication (NEXUS)** – Provides communication services, network unicast and multicast.
- **Security (GSI)** – Provides authentication and related security services.
- **Information (MDS)** – Distributes access to structure and state information.

- **Health and status (HBM)** – Monitoring of health and status of system components.
- **Remote data access (GASS)** – Remote access to data via various interfaces.
- **Executable Management (GEM)** – Constructing, caching and location of executables

3.3.1 Resource Management (GRAM)

The Globus Resource Allocation Manager (GRAM) is responsible for managing computational resources in a Globus based Grid. A Globus based Grid utilizes a hierarchy of GRAM components. A single GRAM is responsible for a set of resources under the same site-specific allocation policy (single Grid site). Site-specific GRAMs are coordinated by other higher level GRAM components, these components are referred to as "resource co-allocators". GRAM is responsible of resource allocation and process management.

A resource co-allocator component sits on a higher level and coordinates all the lower level GRAMs. GRAM can currently interact with a number of local resource management tools available in a variety of operating systems. The management of memory, storage, networks, and other resources is clearly important, but is not supported in current versions of the Globus Toolkit (Foster, Kesselman, 1998b).

Resource Specification Language (RSL) is a generic language used by Globus to allow scripting of custom computation requests to the Grid. RSL allows the requestor to specify the types of resources needed to execute the job. Processing requests are submitted to the resource co-allocator in a RSL format. The co-allocator interprets an RSL request, and breaks it down into generic requirements. The request is then passed to GRAMs that match the resource type requested, or to compatible resources that are able to execute the request. Once the job has completed, the co-allocator receives the outputs from all the GRAMs involved in the execution, and formulates a coherent return to the requestor.

3.3.2 Communication (NEXUS)

Communication services within the Globus Toolkit are provided by the NEXUS communication library. NEXUS provides a set of communication protocols relevant to a Grid implementation. It provides low-level communication APIs that support a wide-range of higher level communication libraries and languages, such as Remote Procedure Call (RPC). NEXUS communication services are used extensively in the implementation of other Globus modules.

Grid communication needs are diverse. They range from point-to-point message passing, to unreliable multi-cast communications. It is the view of the GGF (Global Grid Forum) that TCP is an inappropriate communication technology in Grid environments. This is due to TCP's high overhead and lack of lower level control.

Traditional high-performance computing interfaces and protocols do not provide the communication abstraction Grids require, hence the definition of NEXUS. NEXUS is designed to support a wide range of lower level communication protocols, but still provide a degree of higher level control over communications. To meet the requirements of widely distributed heterogeneous environments.

3.3.3 Security (GSI)

Globus Security Infrastructure (GSI) provides a set of standard security services. GSI provides Grid participants with a common method of authentication and authorization, utilizing a public key infrastructure (PKI), implemented using X.509 digital certificates. The merits of X.509 and other common Grid security mechanisms are discussed in more detail in chapter 7.

Security requirements in distributed Grid environments include: authentication, authorization, privacy, and other security concerns (Foster, Kesselman, 1998b). It is difficult to adequately address all security requirements of Grid resources, due to the heterogeneous nature of Grids, and the fluidic relationships between its participants and resources.

3.3.4 Information (MDS)

Globus Meta-computing Directory Services (MDS) provides Globus implementations with a rich collection of information about Grid components and resources. MDS stores and makes accessible information such as, architecture type, operating system, versioning information, memory available on machines, network bandwidth and latency, available communication protocols, and mapping of IPs to network technology

MDS provides a suit of tools and APIs for discovering, publishing, and accessing information about the structure and state of a Grid; Lightweight Directory Access Protocol (LDAP) is used to store resource information.

3.3.5 Health and Status (HBM)

Heartbeat monitor (HBM) is a simple service used to remotely monitor the health and status of distributed processes. HBM consists of two components, a client interface and a data-collector API.

The client interface allows a process to register with the HBM on execution; once the client has registered with HBM it then regularly sends “hart beats” to the HBM. If heart beasts are not received from a process, the HBM attempts to determine if the problem exists with the process, or the underlying infrastructure (network, computer, etc).

The collector API allows other process to collect health information about a particular registered process.

3.3.6 Remote Data Access (GASS)

Global Access to Secondary Storage (GASS) is a simple module within the Globus Toolkit that provides remote access to files. GASS sub-system allows programs to use standard C I\O library to open, read\write, and append to files stored on remote computers.

3.3.7 Executable Management (GEM)

Globus Execution Management (GEM) supports the execution of process within the Globus Toolkit. GEM supports the identification of suitable locations to execute the desired process, within heterogeneous environments. GEM provides mechanisms for matching hardware required to the executing runtimes requirements

3.4 Conclusion

This chapter introduced the Open Grid Service Architecture (OGSA) as a specification for building largely distributed Grids. OGSA defines a set of Grid requirements and capabilities; these requirements and capabilities were discussed in detail. The OGSA version 1.5 specification (Foster, et al, 2006) formed the basis for this discussion. A primary requirement for OGSA based Grid systems is to provide interoperability between heterogeneous environments. The OGSA specification builds on several technologies from the Web-service community, to provide an interoperable and scalable service orientated architecture. (Further, the key to providing a stateful service orientated framework in OGSA based Grid systems, is the Web-service resource framework (WS-RF), which will be discussed in the next chapter). The Globus Toolkit - in its fourth revision and often abbreviated as GT4 - was also discussed, as an open source software development project implementing OGSA concepts.

Chapter 4

WS-RF, Service Orientated Architecture and Grid Messaging

Chapter 3 introduced the Open Grid Service Architecture (OGSA). OGSA is a specification for building largely distributed heterogeneous Grids. As noted, Grids are built on a Web-service based Service Orientated Architecture (SOA), and provide interoperability between heterogeneous systems (Gerndt, 2004). Core to providing a service orientated framework in OGSA based Grid systems, is the Web-service resource framework (WS-RF). Other key Web-service technologies utilized in Grid environments include: Web Service Description Language (WDSL), Simple Object Access Protocol (SOAP), and Extensible Markup Language (XML). These technologies form the basis for all Web-service technologies. In order to effectively define a Grid security framework, one must have an understanding of how these technologies operate. This chapter will discuss the Web-service based Service Orientated Architecture (SOA), and how it is implemented in a Grid environment.

Schulze and Madeira (1997) define Service Orientated Architecture (SOA) as an architecture that "supports the service lifecycle tasks of development, deployment, hosting and registration, and discovery and invocation". SOA outlines two basic roles, the provider and the consumer. The provider develops, deploys, hosts, registers and manages the service, while consumers discover, and uses these services (Brebner, Emmerich, 2005). Within a Grid context, providers develop and publish services, then provide mechanisms for consumers to discover and consume the service. Web Service Description Language (WSDL) is used to advertise the published service to consumer populations, while Web-service resource framework (WS-RF) defines mechanisms for consumers to access and consume services in a stateful manner.

WS-Resource Framework is defined as, "a set of six Web services specifications that define what is termed the *WS-Resource approach* to modeling

and managing state in a Web services context" (Czajkowski, Ferguson, Foster, Frey, Graham, Sedukhin, Snelling, Tuecke, Vambenepe, 2004b). The WS-resource framework version 1.1 whitepaper defines five Web-service specifications. A sixth standard, WS-notification was added at a later date (March 5, 2004). The WS-resource approach provides a means to express relationships between stateful resources and Web-services (Ibid). This is achieved through six supporting Web-service specifications, these specifications include: WS-ResourceLifetime, WS-ResourceProperties, WS-RenewableReference, WS-ServiceGroup, WS-BaseFaults, and WS-BaseNotification. Each of these specifications will be briefly discussed in this chapter, and how they are utilized in a Grid context to provide a SOA. WSDL and SOAP will be discussed in more detail as well. WSDL is used for providing resource publishing and discovery. It is a document written in XML and is used to describe a Web-service. SOAP facilitates message exchanges between Web-service providers and consumers.

4.1 Service Orientated Architecture

Grids are built on a Service Orientated Architecture (SOA). In order to define a Grid security framework it is important to understand how SOA works. The Web-service SOA provides standardization for interoperability between heterogeneous systems participating in a virtual organization.

SOA is an evolution of traditional client server interactions. SOA outlines two basic roles: a provider and consumer, similar to traditional client\server service model. However, SOA provides a publishing and discovery services in order to facilitate consumer and provider interactions. These interactions can occur without the two parties having any prior knowledge of each other (no configuration required). This flexibility is ideal for Grids, due to the dynamic nature of Grid environments.

Providers or *Grid resources* within a Grid context provide a service to a community of Grid consumers. Consumers access and utilize services published by providers. A single Grid site, or participating organization might contain a number of Grid resources, and Grid users. Grids implement a SOA utilizing Web-services

and related standards, “Web-services standardize the messages that entities in a distributed system must exchange in order to perform various operations” (Open Grid Forum, 2005).

OGSA Grids utilize the Web-Service resource framework (WS-RF) to provide a SOA. WS-RF makes provision for publishing and discovery of stateful Web-services. The following section will discuss WS-RF in more detail.

4.2 Web-service Resource Framework (WS-RF)

WS-RF is addresses short-comings in the Open Grid Service Infrastructure (OGSI), the precursor to OGSA. WS-RF defines a set of conventions and extensions on the use of Web Service Definition Language (WSDL) and XML Schema to enable stateful Web services (Czajkowski, Ferguson, Foster, Frey, Graham, Maguire, Snelling, Tuecke, 2004). Older versions of the OGSA specification were conceptualized around OGSI. OGSI was found to be complex and not easily implemented. WS-RF is an evolution of OGSI, and brings the convergence of the Web-service and Grid communities. It is a Web-service based standard that has been developed concurrently with OGSA. WS-RF allows WS-Resources to be declared, created, accessed, monitored for change, and destroyed via conventional Web services mechanisms (Ibid).

The WS-RF is made up of six technical specifications which define the WS-Resource approach, in terms of specific Web-service message exchanges and related XML definitions. Liming describes the relationship between WS-RF and XML as follows, “WS-RF specifies how XML can be used to describe and access resource properties, clarifies how stateful resources are addressed, and defines how resources may be created or destroyed, individually or collectively” (Liming, Garritano, Tuecke, 2004). The following table describes each of the standards that make up the WS-RF, and their general purpose within the framework:

Table 4.1: A list of WS-RF specifications

Name	Description
WS-ResourceLifetime	Addresses three important aspects of lifetime management, creation of a resource instance, identification, and destruction.
WS-ResourceProperties	Provides a definition of a WS-Resource, in terms of the resource properties. Also provides mechanisms for retrieving, changing and deleting WS-Resource properties.
WS-RenewableReference	Provides Web-service end point management functions.
WS-ServiceGroup	Defines a means to manage multiple heterogeneous Web-service references.
WS-BaseFaults	A base fault XML type used to return error and exception information.
WS-Notification	This is a separate set of specifications, which builds on the WS-RF. WS-Notification provides a system for publisher and subscriber interactions between Web-services and users.

The following sub sections will discuss these specifications in more detail.

4.2.1 WS-ResourceLifetime

This specification is primarily concerned with lifetime management issues around the invocation of a Web-service. The WS-ResourceLifetime specification addresses three important aspects of the WS-Resource lifecycle, these include: creation, management and destruction.

New WS-Resources are created through a WS-Resource Factory. WS-Resource Factory is based on a commonly used pattern for object creation (Gamma, Helm, Johnson, Vlissides, 1995). A WS-Resource Factory is defined as, “any Web-service capable of bringing one or more WS-Resources into existence” (Czajkowski, et al, 2004). The typical result of WS-Resource factory is at least one endpoint reference to a new WS-Resource. Stateful resource information is encapsulated within the WS-Resource implementation. The WS-Resource stateful resources are identified through the use of a stateful resource identifier. Czajkowshi explains, “The form and contents of the stateful resource identifier carried in the reference properties is completely encapsulated within the WS-Resource

implementation" (Ibid). WS-Resources are intended to provide the ability to retrieve a resource identity. The identity should be portable.

Typically only the requestor for a WS-Resource from a WS-Resource factory will be the only interested party in that resource, at least for some finite period. After that period has expired, it should be possible to destroy the WS-Resource, in order to claim back the system resources used in its creation and period of existence. WS-RF standardizes two approaches for the destruction of WS-resources:

- Immediate, and
- Scheduled destruction

Immediate destruction of a WS-Resource might be necessary for any reason. This often achieved by the requestor to send the appropriate request to the resource. *Scheduled* destruction allows for WS-Resources to be destroyed at a later stage, a number of possible reasons. The requestor might not wish to destroy the resource, or might be unable to do so.

4.2.2 WS-ResourceProperties

The WS-ResourceProperties specification defines a method for service requestors to view and modify the state of a WS-Resource's state. WS-ResourceProperties relies on the following three ideas to perform its task (Ibid):

- Each WS-Resource has an XML *resource property document* defined using an XML scheme;
- Service requestors may determine the WS-resource's type by retrieving the WSDL (Web Service Description Language) portType definition, via standard Web-service means;
- And a service requestor may use Web-service message exchanges to read, modify, and query the XML document representing the WS-Resource's state.

4.2.2.1 WS-Resource Properties Document

The WS-Resource properties document acts as a view of the state of the WS-Resource, expressed in XML. Service requestors can request the properties document. The document defines the structure service-requestor-initiated query and update messages can be directed. Consider the following scenario, as described in the WS-RF whitepaper (Ibid): Consider a stateful resource named "C." If the state of "C" comprises three resource property components, named p1, p2, and p3, then its resource properties document, named "ExampleResourceProperties," might be defined as follows.

```
<xs:schema
  targetNamespace="http://example.com/ResourcePropertiesExample"
  xmlns:tns="http://example.com/ResourcePropertiesExample"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
…
... >

  <xs:element name="p1" type= … />
  <xs:element name="p2" type= …/>
  <xs:element name="p3" type= … />

  <xs:element name="ExampleResourceProperties">
    <xs:complexType>
       <xs:sequence>
         <xs:element ref="tns:p1" />
         <xs:element ref="tns:p2" />
         <xs:element ref="tns:p3" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
…
</xs:schema>
```

Figure 4.1: An example of a simple WS-Resource properties document

A service requestor can obtain and view this document through various means. The service requestor learns of the Global Element Declaration (GED) named "ExampleResourceProperties" from the WSDL (Web Service Description Language) portType definition of the Web service component of the WS-Resource.

The WS-Resource properties document declaration is associated with the WSDL portType definition via the use of the ResourceProperties attribute, as in the following example (Ibid):

```
<wsdl:definitions
  targetNamespace="http://example.com/ResourcePropertiesExample"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsrp=
   "http://www.ibm.com/xmlns/stdwip/web-services/ws-resourceProperties"
  xmlns:tns="http://example.com/ResourcePropertiesExample"
…>
…
  <wsdl:types>
    <xs:schema>
      <xs:import
        namespace="http://example.com/ResourcePropertiesExample"
        schemaLocation="…"/>
   </xs:schema>
  </wsdl:types>
…

  <wsdl:portType name="SomePortTypeName"
    wsrp:ResourceProperties="tns:ExampleResourceProperties" >
    <operation name="…
…
  </wsdl:portType>
…
</wsdl:definitions>
```

Figure 4.2: An example of a WSDL portType definition

This association between the portType and resource properties document effectively defines the type of the WS-Resource.

4.2.2.2 WS-Resource Property Composition

In WSDL 1.1, the designer of a Web-service interface composes the interface of the operations defined in the constituent portTypes used in the composition. A portType can constitute multiple standards and specifications to produce a final, complete set of message exchanges to be implemented by a Web service.

4.2.2.3 Accessing WS-Resource Property Values

The state of a WS-Resource, i.e., the values of resource properties exposed in the WS-Resource's resource properties document, can be read, modified, and queried by using standard Web services messages (Ibid).

4.2.3 WS-RenewableReference

WS-Renewable Reference renews a Web-service endpoint that becomes invalid. These mechanisms can be useful to WS-Resource endpoints as they can provide persistent and stable reference to the WS-Resource that can allow the same state to be accessed repeatedly over time.

4.2.4 WS-ServiceGroup

The WS-Service group specification is used to manage multiple WS-Resources.

4.2.5 WS-BaseFaults

The WS-BaseFaults specification defines a base fault type. Base fault types are use for returning fault information when an error occurs during a Web-services message exchange. WS-BaseFaults is used by all of the other WS-RF specifications, to provide consistent reporting of faults relating to WS-Resource definition and use (Ibid).

4.2.6 WS-Notification

The WS-Notification specification is separate to the core WS-RF specifications. WS-Notification defines a Web-service system for publisher\subscriber interactions (Ibid). The specification builds onto WS-RF to provide notifications to subscribers on a ‘topic’ of interest, such as resource property value changes for a WS-Resource. WS-Notification essentially builds on the utility of WS-Resource by allowing requestors to ask to be asynchronously notified of changes to resource property values.

4.3 Grid Messaging

It was previously discussed Grids are implemented utilizing a Web-service based Service Orientated Architecture (SOA). Grid SOA is implemented through the Web-service resource framework (WS-RF) set of specifications. WS-RF provides Web-services with publishing, discovery, as well as state management services. Web-services differ from other distributed computing paradigms. This is due to its focus on XML based Web-standards to address heterogeneous distributed computing (Foster, et al, 2002a). WSDL and SOAP are two XML based standards, which provide a platform neutral message exchange mechanism. These mechanisms allow for Grids to support the dynamic discovery and composition of services in heterogeneous Grid environments. WSDL has a focus on describing services, while SOAP is more concerned with facilitating communication.

4.3.1 Web-service Description Language (WSDL)

Web Service Description Language (WSDL) is a core technology in Grid implementations. Foster describes WSDL within an OGSA context as follows:

- "This architecture (OGSA) uses the Web Services Description Language (WSDL) to achieve self-describing, discoverable services and interoperable protocols, with extensions to support multiple coordinated interfaces and change management" (Foster, et al, 2002a).

Understanding the basic purpose and mechanisms of WSDL will allow for a greater understanding of Grid security. WSDL is defined as an, "XML document for describing Web services as a set of endpoints operating on messages containing either document-orientated messaging, or RPC payloads" (Christensen, Curbera, Meredith, Weerawarana, 2001). Primary in a Grid context, WSDL utilizes a document-oriented messaging scheme, making use of XML documents.

WSDL provides dynamic discovery and composition of services in heterogeneous environments necessitates mechanisms for registering and discovering interface definitions and endpoint implementation descriptions. WSDL supports this requirement by providing a standard mechanism for defining interface

definitions separately from their embodiment within a particular binding (transport protocol and data encoding format) (Foster, et al, 2002a). WSDL enables the publishing of services across multiple network protocols and message encoding formats.

WSDL defines a Web-service as collections of communication end points that can exchange certain messages. WSDL documents describe a Web-services interface and provide users with a point of contact on the remote server. In other words a WSDL document will describe an abstract interface for remote users to connect to, and specific protocol-dependent details that users must follow to access the service.

4.3.2 Simple Object Access Protocol (SOAP)

Simple Object Access Protocol (SOAP) is a simple enveloping mechanism for XML, and provides a means of messaging between a service provider and requestors. SOAP is an XML-based protocol for messaging and remote procedure calls (RPCs). It provides a platform independent and lightweight communication protocol over the World Wide Web (WWW) (Curbera, Duftler, Khalaf, Nagy, Mukhi, Weerawarana, 2002). On top of a basic messaging structure, the SOAP specification defines a model that dictates how recipients should process SOAP messages.

SOAP documents are typically a simple XML document with a single element and two child elements (Ibid). The first element is typically a header and the second contains body elements. Consider the following figure, a basic SOAP envelope:

```
<SOAP:Envelope xmlns:SOAP=
        "http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
        <!-- content of header goes here -->
    </SOAP:Header>
    <SOAP:Body>
        <!-- content of body goes here -->
    </SOAP:Body>
</SOAP:Envelope>
```

Figure 4.3: A simple SOAP envelope

SOAP can be used to execute Remote Procedure Calls (RPC) on the desired Web-service. In order to utilize SOAP to execute remote functionality, standard mechanisms are required on the server to transform the SOAP XML representation of variables and call data into native typed values. The World Wide Web Consortium (W3C) provides standard XML scheme specification for providing a standard language for defining the document structure and the XML structures' data types.

Consider the following scenario, a user wishes to execute a remote procedure to do some arbitrary unit of work. The user learns of the service and its requirements via WSDL services. The desired service has a function name of, "UserFuntionA", this function takes a single integer type as a parameter, and returns a two-row data structure. A SOAP document is passed to the service, via a standard HTTP POST command. The document is parsed and interpreted by the Web server. In this SOAP envelope, the call to "UserFuntionA" is an XML element with attributes that include information about the encoding (note the references to XML, "http://Schemas.xmlsoap.org/soap/envelope"). The element's children are the method call's arguments, in this instance the integer value taken in as a parameter, by the function. Once the operation is complete the service returns a formatted XML document to the service caller, with the desired output. To make the above interaction possible both parties must agree on the XML scheme for communications.

4.4 Conclusion

This chapter discusses Service Orientated Architecture (SOA), and how it relates to the implementation of OGSA compatible Grids. The Web-Service Resource Framework (WS-RF) is a set of specifications utilized by OGSA to provide stateful Web-services. Web-services are built upon to provide communication between heterogeneous and dispersed parties. Two standards that support WS-RF were discussed: Web Service Description Language (WSDL) and Simple Object Access Protocol (SOAP). WSDL is a method to describe services, while SOAP is primarily concerned with simple, low overhead communication.

Chapter 5

STRIDE and OGSA Grid Layers

Chapter 4 discussed the WS-RF. WS-RF is utilized to implement a Web service-based SOA within OGSA-based Grids. The goal of this research is to work towards a holistic framework for OGSA-based Grid security. Providing adequate security to Grid users will allow for wide user adoption of the technology (Schopf, 2002).

The OGSA specification defines capabilities that can be represented in three high-level tiers (Foster, et al, 2006); these tiers are a logical, abstract, semi-layered representation of some of the OGSA capabilities. In the article entitled "Anatomy of the Grid" (Foster, et al, 2001) discusses a five-layer structure, outlining technical requirements at the various OGSA capabilities tiers. Some of these layers overlap between OGSA tiers.

For the purpose of a Grid-security framework, this research will outline two logical layers. These layers are derived from previous work done by Foster and other contributors to the definition of the OGSA specification (Foster, et al, 2001, 2006). These two layers are the political (inter-organisational) "Common Grid Layer" and the "Local Grid Resource Layer". The logical division is made for the purpose of identifying security challenges faced by Grid designers. Both layers have their own unique set of security challenges. In order to define an effective holistic security strategy; all or most of the security challenges faced in each tier need to be identified and addressed, if possible.

The *Common Grid Layer* is primarily concerned with the interaction of all participating sites in a virtual organization. The challenges faced on this layer are typically cross-organization trust, and securely administering site-to-site interactions. Virtual organizations are constantly-changing, dynamic structures. There are many physical organizations participating within a virtual organization context, having varying levels of trust between them, as well as varying and potentially incompatible security policies and technologies. The challenge for Grid designers on the common Grid layer is to provide transparent cross-site

authentication and authorization. When a user authenticates to a Grid he\she should be able to use any resource, provided they have the correct access privileges, without being required to constantly authenticate.

The *Local Resource Layer* is concerned with data and computational Grid resources as a separate local entity below the Common Grid Layer. This layer is concerned with the lower-level Grid security issues. Grid resource protocols (OGSA middle tier) are concerned entirely with Grid fabric, and hence ignore issues of global state and atomic actions of the distributed collections (Foster, et al, 2001). For this reason one can focus on local Grid resource issues. Some of the primary security concerns for Grids on this layer include: the hosting environment and applications security, machine and operating system security, network and communication security, and message security (See Grid specific messaging, typically XML based, in chapter 4) (Siebenlist, Welch, Tuecke, Foster, Nagaratnam, Janson, Dayka, Nadalin, 2002).

The goal of Grid computing is to provide resource sharing, whether it is storage, computational power, or specialized hardware; in a non-platform\hardware specific manner, promoting collaboration between heterogeneous, globally dispersed parties (Foster, et al, 2001). The possibility of defining a security strategy, considering each possible attack avenue that attackers can make use of, on each commercial and non-commercial operating system, hosting environment, multiple possible network and communication protocols, etc., is overwhelming. When considering the possible magnitude of a Grid implementation, within a virtual organization context (VO), the task seems more then a little onerous. One feasible approach – as advocated in this study - is use a threat-perspective, to group attacker's goals and action into generic categories, for the purpose of defining a Grid-security framework.

To this end, the security framework defined in this research will be based on STRIDE. STRIDE is an acronym for prominent threat categories faced by computer systems and can be extrapolated to include Grid computing. These threats are grouped into six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service, and Elevation of Privileges (Meier, et al, 2003). The

merits of STRIDE will be discussed in this Chapter. STRIDE as a threat classification scheme will be utilized in Chapter 6 and Chapter 7.

The following Chapter will highlight the security challenges faced at each layer in more detail, and propose possible solutions to be considered for a holistic framework for Grid security.

5.1. Grid Security

In a survey published by Schopf, it was found that in order for Grids to achieve wide adoption, they must be secure enough (Schopf, 2002). Grids need to assure secure access and communication. Due to Grid requirements and the nature of Grids, these requirements are not easily met. Grids introduce a new set of security challenges; a Grid will, typically, make use of multiple communications and transport protocols cross organization and administrative domains and be deployed across multiple platforms and operating systems (Foster, et al, 2001; Foster, et al, 2002a). Grid-security issues can be logically divided into two levels. There are issues around underlying technologies and infrastructure (lower Grid level), as well as higher-level political issues (higher-Grid level).

Lower-level Grid-security issues are primarily concerned with platform and OS security, application security, and network security (*The Local Resource Layer*), while higher-level Grid issues include: authentication, authorization, accounting, credential delegation and conversion, and single sign-on (*The Common Grid Layer*).

The goals of this research are two-fold, (1) to define a broker to facilitate high- level Grid security needs (Chapter 6, Common Grid layer), and (2) to define a generic security strategy Grid designers can implement to secure their Grid fabric (Chapter 7, Local Resource Layer). The STRIDE threat classification scheme will be used to evaluate the broker defined in Chapter 6, while in Chapter 7 STRIDE will form the basis of the proposed Grid resource security framework. The security framework will be defined from the perspective of a Grid; and how its components are at risk of attack, based on the goals or intentions of an attacker. (STRIDE categories will be primarily considered here). This approach is known as threat-

modeling (Heckman, 2006). The threat-modeling process allows Grid security architects to methodically identify threats faced by their Grids, providing assurance to Grid users and maintaining information security.

5.1.1 Information Security

Information Security is concerned with protecting information and computing resources from external threats and attacks (Whitman, Mattord, 2003, pg 9). A threat is the possibility of a Grid asset being attacked or compromised, via a vulnerability that exists in the Grid. Vulnerabilities are weakness in systems (Grids) that are exploited by attackers when attacking an asset. In order to minimize security failure, one must have an idea of what security success entails (Hernan, Lambert, Ostwald, Shostack, 2006). Hernan et al explain that a secure system has the properties of confidentiality, integrity, and availability and that users are authenticated and authorized correctly, and that transactions are non-repudiable (Ibid). The three properties of information security, i.e. confidentiality, integrity, and availability, are collectively known as the CIA triangle.

The CIA triangle has been considered an industry standard for information security since the development of the mainframe (Whitman, Mattord, 2003, pg 10). The CIA triangle is still considered widely relevant today in the information-security field. CIA properties are defined as (Hernan, et al, 2006; Whitman, Mattord, 2003, pg 10-13):

- **Confidentiality:** Data is only available to the people intended to access it.
- **Integrity:** Data and system resources are only changed in appropriate ways by appropriate people.
- **Availability:** Systems are ready when needed and perform acceptably.

Threat-modeling is a method that can be employed by Grid designers, to ensure that Grids have these security properties (Ibid)

5.1.2 STRIDE As A Threat-classification Scheme

There are a wide range of possible attacks, and further fine-grained variations on these attacks. The best method to classify threats to one's system is to identify the hacker's goals when performing an attack. STRIDE is relevant to Grids and will be used to evaluate the proposed brokered approach in Chapter 6, and will be used as the basis for a Grid resource security framework in Chapter 7. As mentioned, STRIDE is an acronym used to group the following types of threats (Meier, et al, 2003; Hernan, et al, 2006):

- **Spoofing** - The hacker's goal when spoofing is to try gain access to the system by mimicking legitimate user-credentials or network traffic.
- **Tampering** – This is the unauthorized altering of information, while it is in transit between two computers.
- **Repudiation** – Prevents administrators from knowing if users (legitimate or not), have performed an action.
- **Information disclosure** – This is the unwanted exposure of private information.
- **Denial of Service** – This is the process of making services unavailable to users.
- **Elevation of privileges –** This attack occurs when a user of limited privileges assumes the roll of a privileged user, in order to steal, corrupt, or deny access to information asset.

The following sub-sections will discuss these threat categories in more detail.

5.1.2.1 Spoofing

Spoofing is when a hacker tries to gain access to system illegitimately, by mimicking legitimate behaviour. Typical forms of spoofing applicable to Grids are user-credential spoofing and IP-address spoofing.

User-credential spoofing occurs when an illegitimate user obtains a legitimate user credential or certificate. This is typically through some method of password guessing, such as brute force, or dictionary attacks. User-credential

spoofing allows a hacker to "walk in the front door", allowing them to bypass security measures put in place.

IP-address spoofing occurs when a hacker assumes the TCP\IP (Transmission Control Protocol\Internet Protocol) address for the purpose of exploiting a trust relationship between communicating parties. This method of spoofing is harder to achieve than user-credential spoofing, and as a result is less common practice.

5.1.2.2 Tampering

Tampering is the unauthorized altering of information, either when in transit between two communicating parties, or when stored on a network terminal. The goal of the attacker when tampering could be either to gain access to a Grid, by altering network traffic into fooling authentication mechanisms, or it could be to destroy and invalidate data stored on network servers or terminals.

5.1.2.3 Repudiation (Non-Repudiation)

Repudiation is when the actions of users cannot be verified, typically through system logs being deleted.

5.1.2.4 Information Disclosure

Information disclosure is the unwanted exposure of private information.

5.1.2.5 Denial-of-Service (DoS)

Denial of service is the process of making computing services unavailable to the users. Denial of service attacks are often the last resort for an attacker that cannot successfully penetrate a Grid and launch their desired attack.

5.1.2.6 Elevation Of Privileges

This occurs when a user with limited access credentials is upgraded to a user with greater access credentials, and is a typical method used by attackers to launch an attack against a Grid.

5.1.3 STRIDE and Grid Threat-modelling

Threat-modelling is defined as, "the methodical review of a system design or architecture to discover and correct design-level security problems" (Hernan, et al, 2006). STRIDE is considered to be a finer-grain version of the CIA triangle (Howerd, Lipner, 2003). STRIDE was discussed in the previous section. From a threat-modelling perspective it is possible to map threats defined by STRIDE to system security properties beyond the CIA triangle. Consider the following table (Hernan, et al, 2006):

Table 5.1: A mapping of STIDE threat categories to security services

Threat	**Security Service**
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Denial of Service	Availability
Elevation of privileges	Authorization

From the above table it can be seen that STRIDE defines an additional three security services to those defined by the CIA triangle. These services are: authentication, non-repudiation, and authorization. Consider the following definitions of these services (Hernan, et al, 2006). (Refer to section 5.1.1 for definitions of CIA properties).

- **Authentication:** The identity of users is established (some challenge mechanism is require, i.e. username and password).
- **Non-repudiation:** Users cannot perform an action and later deny performing it.
- **Authorization:** Users are explicitly allowed or denied access to resources, based on their authentication credentials tested against a policy.

These additional services build on the CIA in that they provide consideration for modern networked systems. Since Grids are primarily network-based systems, this makes STRIDE a strong foundation for a Grid-security framework.

An attacker will generally employ several generic steps to attack a system (See Chapter 7, section 7.1.1 for more details). Blocking multiple steps or employing multiple ways to stop one step is considered a defence in-depth approach to security (Salter, 1998). STRIDE threat categories and their related security services allow one to address different aspects of an attack. This will be discussed in more detail in Chapter 7.

5.1.4 Conclusion

This section discussed information security principles. The CIA triangle was discussed as well as STRIDE. STRIDE defines an additional three security services to the CIA triangle. The following sections will discuss OGSA Grid physiology, as defined by Foster (Foster, et al, 2001; Foster, et al, 2006). The two logical Grid layers mentioned earlier will be defined and discussed. These layers will be used as a foundation for defining a Grid-security framework. The following sections will discuss the OGSA Grid physiology, defined in the OGSA specification (Foster, et al, 2006).

5.2. OGSA Grid Physiology

OGSA describes three logical tiers in a Grid. These tiers were discussed in chapter 3; they include (Foster, et al, 2006):

The *base resource tier* contains base resources. Base resources are supporting underlying entities and artifacts that may be logical or physical, and have relevance outside OGSA. This may include hardware (CPU, memory, disk space), or OS processes, etc. All of these need to be protected during access from VO members and executing processes, particularly if spoofing, tampering, repudiation, information disclosure, denial of service, or elevation of privilege (STRIDE) attempts are being made.

The *virtualization and abstraction tier* defines capabilities directly relevant to OGSA Grids (See Chapter 3, section 3.2). These capabilities allow for support of applications and processes on a higher level of the Grid architecture. A detailed relationship between the middle tier and lower tier (or base resource tier) exists (Foster, et al, 2006). This tier is where STRIDE attacks need to be addressed in a Grid-wide manner.

The *application tier* is a logical representation of applications and process, built on OGSA to realize user and domain-orientated processes and functionality (such as business processes) (Foster, et al, 2006). All user and process security credentials emanating from this layer need to be brokered between this layer and the base resource layer in order to persist (assert) authentication and authorization (from top to bottom).

Foster describes several additional layers within the three OGSA tiers (Foster, et al, 2001). Each of these layers can logically be mapped to an OGSA-specified tier. These layers include:

- Application layer
- Collective layer
- Resource layer
- Connectivity layer
- Fabric layer

The *fabric layer* consists of all native services and resources (Grid fabric). To reiterate, these are the resources that Grid designers wish to make available to a virtual organizations; this could range from processing components, to highly specialized equipment such as telescopes. As standalone components the elements of the fabric layer are typically built on non-standard platforms, such as JAVA or .NET; and are run on various operating systems, such as Windows, LINUX or UNIX. OGSA-based Grid fabric is exposed to the Grid through open and interoperable Web-service interfaces, thus allowing for SOAP-based security features to be incorporated. Standardised SOAP headers can be used to negotiate security at various levels of the fabric layer (e.g. at network/directory services,

operating system levels/folder-access levels, process-execution/code-object levels, etc).

The *connectivity layer* – which is a sub-set of the middle layer - defines core communication and authentication protocols required for Grid-specific network transactions (Foster, et al, 2001). Grid authentication and authorization is taken care of at this layer. Any network-based security, such as IPSec or TLS, can be applied here. SOAP messaging security typically operates at the application level where SOAP headers are read by processing applications (of services) in the resource layer.

The *resource layer* is where the OGSA specification allows for standalone heterogeneous components, within the fabric layer, to talk to one another transparently and dynamically, as well as logically or grouped components as part of some higher-level workflow or process.

The resource layer builds on connectivity communications and authentication protocols to define protocols (and APIs and SDKs) for secure negotiation, initiation, monitoring, control, accounting and payment of sharing operations on individual resources (Foster, et al, 2001). Protocols at this layer are designed so they can be implemented on top of actual services in the Fabric layer.

The *collective layer* is a logical grouping of resources exposed to the Grid via the fabric layer. This grouping allows for a wide range of global services and application specific behaviours, hence the name collective because it involves the coordinated ("collective") use of multiple resources (Foster, et al, 2001).

The *application layer* is simply the result of all other layers working together to provide non-trivial services within a virtual organization.

The following table illustrates how different aspects of IT infrastructure would be logically grouped in these layers:

Table 5.2: How the layers of a grid are grouped by IT infrastructure

Layers in a Grid	Infrastructure\Service aspects
Application	Any non-trivial Grid derived applications (e.g. cross-organization data warehouse)
Collective	Resource discovery, resource brokering, system monitoring, cross domain authentication
Resource	Access to data, Access to information services (service state, status, etc), performance information
Connectivity	Communication (IP), Discovery Services (DNS, WSDL), authentication, authorization, delegation
Fabric	Processing resources, cluster servers, networks, databases, computers, networks

5.3. Abstract Grid layers

The three high-level abstract tiers of OGSA capabilities were discussed in previous sections. These tiers include the base resource tier (bottom tier), the virtualization and abstraction tier (middle tier), and the application tier (Top tier). In related work, Foster defines five logical layers with regard to Grid capabilities (Foster, et al, 2001). These five layers were discussed in previous sections as well (fabric, connectivity, resource, collective, and application).

These layers are broken down into smaller technical requirements for implementing OGSA-based Grid systems. There is an overlap between these defined layers and the OGSA abstract tiers. For the purpose of this research, Grids will be discussed in terms of two abstracted layers (the Local Resource Layer, and The Common Grid Layer). The Grid is divided into these two layers to address various Grid security challenges.

5.3.1 Common Grid Layer

The common Grid layer is an abstract layer. This layer is concerned with high level Grid security issues. Grids security challenges are unique: Grids typically span organization and administrative domains. Although OGSA defines standard mechanisms for service publishing and discovery, the underlying infrastructure might differ from site-to-site. Sites might implement different authentication technologies (Kerberos, SSH, SSL, etc), might make use of varying platforms and operating systems, incompatible security policies, communication protocols, etc. With these concerns in mind the Common Grid Layer is defined and is concerned with providing Grid users with cross-site authentication and authorization, the delegation of user credentials between various sites, and bridging between incompatible security policies and technologies.

5.3.1 Local Grid Resource Layer

The Local Grid Resource Layer is concerned with the security challenges at a single Grid site. This layer is concerned with securing Grid fabric and Grid resources from outside attackers and threats. A Grid-security framework needs to be generic in order to be widely relevant to Grid participants. Grid participants make use of varying platforms and technologies.
This study will use STRIDE as a basis for defining an overall Grid-security framework.

5.4 Conclusion

STRIDE and the CIA triangle were discussed. It was discovered OGSA Grid capabilities are divided into three high-level tiers. These layers include: the base resource tier (bottom tier), the virtualization and abstraction tier (middle tier), and the application tier (top tier). Furthermore, these layers are a logical, abstract, semi-layered representation of some of the OGSA capabilities. Five more layers (application, collective, resource, connectivity, fabric) within these tiers were discussed; these layers define technical requirements within the OGSA tier

structure. From these layers, two higher levels (the Common Grid Layer and the Local Resource Layer) are derived; these layers are concerned with Grid security challenges. For the purpose of defining a security framework, these two layers will be discussed in greater detail, in Chapter 6, and Chapter 7. Chapter 6 will discuss higher-level Grid security concerns (the Common Grid Layer), while Chapter 7 is concerned with lower-level Grid security concerns (the Local Resource Layer).

Chapter 6

A Brokered Approach To OGSA Grid Security

In chapter 5, two logical layers of a Grid were discussed: the "Common Grid Layer" and the "Local Resource Layer". This section is primarily concerned with the security challenges faced by the Common Grid layer. These challenges include cross-site authentication and authorization, delegation of user credentials, and compatibility of security policies between participants within multi-organizational Grid. This chapter will propose a brokered approach to addressing Grid security issues on the Common Grid Layer. STRIDE was discussed in chapter 5. Elements required to implement a broker will be discussed and evaluated against STRIDE.

The Common Grid Layer is concerned with higher-level Grid security issues. These issues are concerned with the Grid as a whole in a Virtual Organization (VO) context, and all the security challenges associated with cross-domain administration. Grids consist of large dynamic populations that could be made-up of a number of different physical organizations, possibly spanning the globe.

Authentication and authorization between sites could prove to be an administrative challenge. Cross-site authentication maps for every possible user to every possible resource are difficult to implement and maintain. Storing remote user credentials on a local resource is impractical (because of differing authentication mechanisms) and does not allow for scalability. Lingering user credentials for a resource is a major security concern. Virtual organizations potentially have a large user turnover. Users that are no longer part of the VO could have access credentials to Grid resources. Such users are a large security risk.

The Open Grid Service Architecture (OGSA) requires large scale, cross-organization authentication and authorizations (Foster, et al; 2006). Current OGSA implementations (Globus toolkit) primarily make use of X.509 certificates to provide authentication and secure communication between Grid participants.

However this is an extra layer put on top of the existing security infrastructure that exists at participants sites. The goal of this chapter is to:

- Identify the primary security challenges faced by Grid designers on the Common Grid Layer.
- Provide a detailed description of a proposed strategy for implementing managed site-to-site authentication, utilizing a brokered approach.
- Apply STRIDE as a threat classification scheme, and evaluate the proposed brokered approach to it.
- Finally, conclude and reflect on the proposed brokered approach.

6.1 Cross-site Grid Security Challenges In An OGSA Context

Sites participating in Grid, within a virtual organization (VO) context are generally managed and maintained by separate administrative groups. The OGSA specification has no formal definition of a brokered service although Foster describes the use of a broker to simplify inter-Grid communication and trust services (Foster, Kesselman, Tsudik, Tuecke, 1998). Consider the following scenario: a typical Grid interaction between Grid participants as described by Foster (Ibid):

User-A at Site-A starts an analysis program that sends code to be executed on Site-B, but Site-B requires a dataset on Site-C to perform the analysis. The application at Site-A contacts a broker at Site-D to obtain idle resources needed to process the task at hand. The broker then initiates communications with sites E, F, G in order to complete the task at hand. These sites will need to maintain communication between them (possibly using a multicast protocol), as well as the broker, the original site (requesting site), and the user.

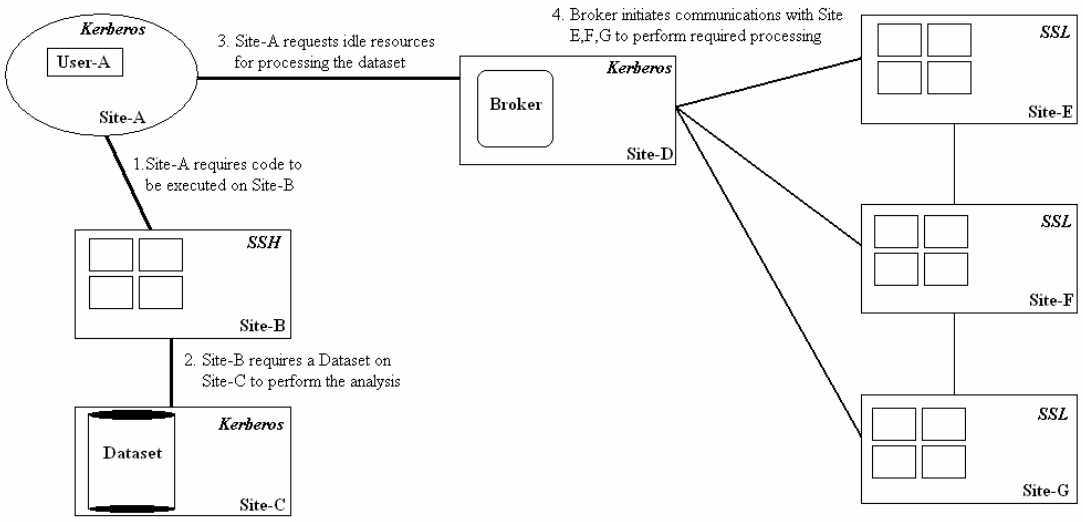

Figure 6.1: Example of a large scale distributed computing environment

The Above scenario (figure 6.1) depicts many distinctive characteristics of a Grid computing environment (Foster, et al, 1998):

- The resource pool is large and dynamic.
- A computation (or processes created by a computation) may acquire, start processes on, and release resources dynamically during its execution.
- The processes constituting a computation may communicate by using a variety of mechanisms. Low-level communications (e.g. TCP/IP sockets) can be created and destroyed dynamically during program execution.
- Resources may require different authentication and authorization mechanisms and policies, which we will have limited ability to change. In the above example, this was illustrated by showing the local access control policies that apply at different sites. These include Kerberos, Secure Socket Library (SSL) and Secure Shell (SSH).
- An individual user will be associated with different local name spaces, credentials, or accounts, at different sites, for the purposes of accounting and access control.
- Resources and users may be located in different countries.

There is need to provide security solutions to Grid users that can allow computations, such as in the above described scenario. These solutions must allow for the co-ordination of diverse access control policies and to allow them to operate securely in heterogeneous environments (Foster, Kesselman, 1998).

The cross-domain Grid administrations must provide Grid participants with the following set of requirements, to allow Grid users to transparently use Grid resources (Butler, Engert, Foster, Kesselman, Tuecke, Volmer, Welch, 2002; Nagaratnam, Janson, Dayka, Nadalin, Siebenlist, Welch, Foster, Tuecke; 2002):

- **Authentication:** Authentication points that support multiple authentication technologies and protocols are required.
- **Authorization:** Authorization should regulate the access to Grid resources based on access-control policies policies.
- **Single sign on:** Users must be able to logon once, and have access to multiple Grid resources without having to constantly provide credentials.
- **Delegation:** A program must have the ability to run on the user that initiates its behalf. This allows the program to access the resources that it might need, that the calling user has access to.
- **User-based trust relationships:** In order to provide transparency to Grid users, Grid site security administrators must have no need to interact with one another. Grid participants should be able to "plug-in" and not have to consciously make provision for inter-site trust. Some middleware is required to automate these tasks.

In order to support a global Grid security infrastructure within a VO, a broker can be used to facilitate communications, authentication and authorization at a central site. A broker could take into consideration all the above mentioned requirements. It can be implemented through the use of various services and middleware. The following section will discuss a broker within an OGSA context.

6.2 Brokered Approach To Interoperable Security

This section is concerned with defining a brokered approach to implementing a uniform Grid-wide security structure. As highlighted in the previous section, Grid designers are faced with the challenge of providing robust and scalable cross-site trust relationships. Sites within a virtual organization often make use of incompatible security technologies, such as Kerberos, Secure Socket Layer (SSL), Secure Shell (SSH), etc. It becomes a tedious administrative task for local administrators to define local access credentials for all external users that wish to access local resources.

In order to make large-scale VOs feasible for mainstream adoption, mechanisms are needed to provide a context to associate users, requests, resources, policies and agreements across operational boundaries. When sharing resources across organizational boundaries, certain security needs are implied (OGSA specification 1.5). The brokered approach could implement many of the Grid user requirements (as identified in section 6.1). These include: Single sign-on, delegation, interaction with various local security solutions, and user-based trust relationships.

6.2.1 The Functions Of A Broker In A Grid

The proposed broker is an abstract software component. Its primary purpose is to provide centrally hosted services to facilitate secure interoperable communications between Grid participants. The broker facilitates the complex trust relationships that exist between Grid parties. When a user initiates a communication to a resource on another site, the broker will handle authentication and authorization of that user. The broker stores a directory of all Grid users and resource, as well as access control information to the Grid resources.

The broker will have unique identification of all users on the Grid, based on a mapping of their "site code" and unique local username combined. The broker will have a unique identification code for each site participating in the Grid. The Grid directory structure is used to store user credentials and authorization

mappings. The broker stores some meta-data about sites in the Grid. This includes security information about a site, what authentication technology it makes use of, i.e. Kerberos, SSH, SSL, etc.

6.2.2 Implementation Of A Broker

The following section will provide more detail on the proposed broker's logical structure. The broker provides elements to allow the single sign-on and delegation of user credentials. A resource can utilize other Grid resources on the user's behalf (delegation). The broker should allow for authentication of users, resources, and processes and must support user-to-resource, resource-to-user, process-to-resource, and process-to-process authentication (Foster, Kesselman, 1998). For the purpose of providing such a complex authentication and authorization strategy, it is proposed the broker makes use of a "group policy" structure, in order to facilitate these needs. The structure consists of several elements:

- Resource
- Users
- Owner
- Groups
- Privilege

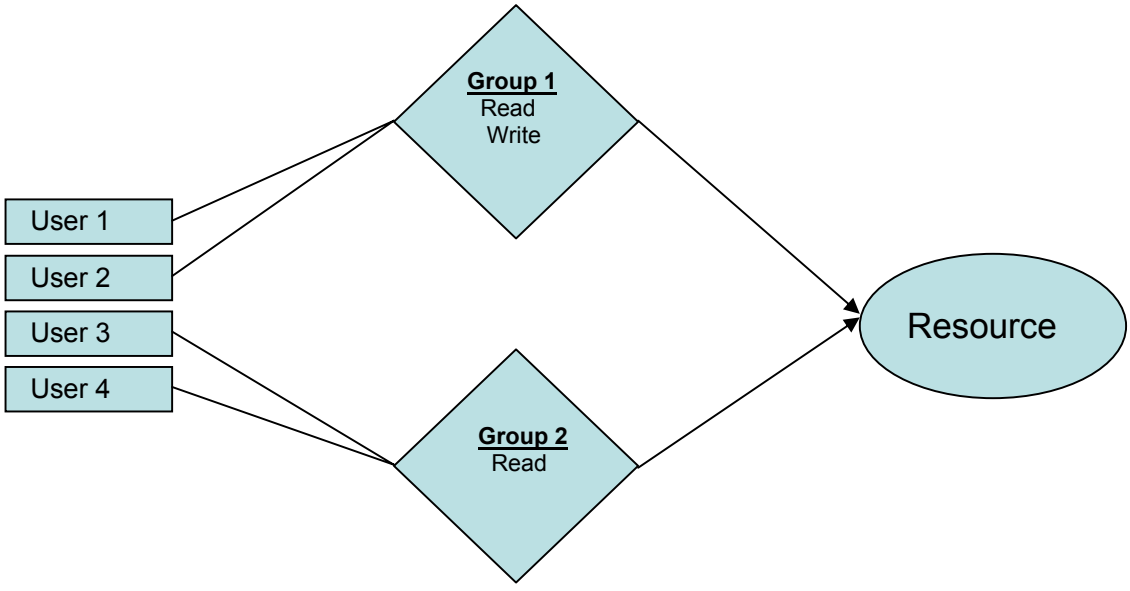

Figure 6.2: Simple diagram of group policy structure

Figure 6.2 is a graphical representation of how users interact with resources. Resources belong to participating sites in the Grid and are managed by their respective administrators. These administrators are defined as resource owners. Groups are defined and linked to a resource. One group can contain many users. A group then has a privilege to access the resource defined, one resource can have many groups linked to it. Only owners can assign access rights to resources. This simple paradigm can allow for complex authorization structures throughout the Grid and can cater for scalability.

The process of defining user access credentials for all users on a Grid that can access a particular resource could be an administrative nightmare and not good practice, in terms of scalability. Within the proposed brokered approach, sites can define a standard set of access credentials or user logons, such as “Read”, “Write”, etc. These credentials can be seen as “local” proxy accounts, managed on the broker side. Access to the resource is monitored and all the accounting will occur at the broker. The broker will determine which local credential to invoke (or proxy to open), based on the requesting users access rights to the resource, which are stored and determined in the broker directory.

A multi-layered administration strategy for the broker is required. Considering access to site resources is determined at the broker; the broker dynamically authenticates users to a resource. Site administrators need some level of control over who can and cannot access their resources, and what degree of access is granted to various parties.

A brokered approach can provide scalable authentication and authorization to Grid environments, but it could also be seen as a central point of failure that could potentially bring down the entire Grid. There are a number of ways to combat this, including:

- Redundant hardware (RAID, clustering, Network Interface Cards [NICs])
- Clustering
- Site Replication

6.3 Elements Of The Grid Broker

The goals of the broker are to provide a higher-level abstracted service for handling complex site-to-site authentication and user-based trust relationships within a Grid. Some requirements for providing a transparent user interaction on a Grid when utilizing resources from multiple sites were identified (Butler, et al, 2002; Nagaratnam, et al, 2002): authentication, authorization, single sign on, delegation, user-based trust relationships. This seeks to identify services a broker would utilize to support these security requirements; and how these services are at risk from attack when considering STRIDE.

The following list provides a proposed set of components and services required by a broker to perform its desired tasks. Each service will be discussed in more detail in a subsequent section:

- **Authentication:** Single sign-on allows a user to authenticate at the broker and use any resource available to him\her without requiring constant re-authentication.
- **Authorization:** Authorization services determine what levels of access (privilege elements) to Grid resources legitimately authenticated users possess.
- **Execution management:** Execution management is an OGSA capability. In order to incorporate the benefits of the broker; execution management can be abstracted into the Common Grid Layer and handled by the broker as well.
- **Scheduling service**: Scheduling services is another OGSA capability. Scheduling services are loosely coupled to execution management services (Foster, et al, 2006). For this reason scheduling services are abstracted to the broker as well. Values in security Group elements can be used for prioritizing access and execution management. This can be policy-based (rules-based) as well.
- **Network communication**: Network communication services are not explicitly defined as part of the broker services, but it is necessary to

consider them. Almost all Grid communications occur over some networked medium. A broker can be used to identify network-transport layer security protocols (such as IPSec and SSL/TLS) in both incoming execution requests (SOAP messages) and in Information Services caches regarding security protocol requirements of resources to be accessed by the requests.

- **Storage:** The broker will be required to store many different types of information. Although storage is not an explicit broker service it must implement directory services (user accounts, security groups, group policies, etc.) from a security. It will also provide information (OGSA Information Services) regarding resources (in XML documents).

6.3.1 Authentication

The broker is responsible for negotiating communications between Grid participants. One of the primary functions of the broker is to provide a standardized method for Grid participants to authenticate to each other. As highlighted in previous sections, Grids are made up of heterogeneous environments. They typically comprise multiple physical organizations and administrative domains. Domains or sites often make use of incompatible technologies and platforms. The Open Grid Service Architecture (OGSA) defines a set of capabilities to allow diverse heterogeneous parties to communicate and share resources. However, there are no adequate solutions available to provide scalable authentication and authorization services which are required to form the basis of user-based trust relationships in a Grid system. This section will focus on how a broker could provide authentication services to heterogeneous parties participating in a Grid, and how the broker addresses the requirement of single sign-on capabilities in a Grid.

Consider the following steps taken by the broker when authorizing a user to access a resource:

1. A user provides the credentials needed to log onto the Grid.
2. The user initiates a process that requires remote Grid resources.
3. The user's Grid credentials are tested against the resource's global access-control policy.

4. The user's rights to that resource are determined.
5. If the user has sufficient rights, the Grid initiates the communication and provides that Grid resource with the correct level authentication.
6. Broker passes the WS reference to the client, the interaction then becomes a direct interaction between the user and resource.

A broker is intended to facilitate authentication services between heterogeneous parties. In previous sections (Section 6.1), several high-level Grid security needs were identified. A broker addresses the need for single sign-on. Once a user authenticates to the Grid (broker), the broker will maintain the equivalent of a "ticket" (as implemented in Kerberos) for that particular user. A ticket may be valid for a pre-configured period, i.e., 24 hours. The broker can then use the ticket on the user's behalf to execute services required to complete a computation. This will allow the user to log onto the broker once, and not be prompted to resubmit his\her credentials every time a new resource is required to complete a task. However, for authentication to persist to the fabric level, the user or process's credentials must be translated to that required by the fabric resources. The broker is able to affect this as a result of the knowledge-base possessed by the broker. GSI implements a system of proxy certificates, the function of which can be subsumed by a broker.

6.3.2 Authorization

Authorization as a security service within a Grid environment is defined as, "A service that evaluates policy rules regarding the decision to allow the attempted actions, based on information about the requestor (identity, attributes, etc.), the target identity, policy, attributes, etc.), and details of the request" (Welch, Siebenlist, Foster, Bresnahan, Czajkowski, Gawor, Kesselman, Meder, Pearlman, Tuecke, 2003). Within the context of this proposed Grid broker, a service requestor (Grid participant) will request the use of a Grid service. The broker will determine if the requestor can access the service based on the requestor's defined rights to the service (access policy). The user's rights to a resource are defined by the resource owner (target).

Current OGSA-based Grid implementations, such as the Globus toolkit, make use of Public Key Infrastructure (PKI) to implement authorization assertions between Grid participants (Welch, et al, 2003). The proposed broker is not intended to replace the current implementations of Grid security. The broker merely abstracts this functionality to a higher level. This allows for greater flexibility and scalability in defining access control policies to Grid resources in a dynamic, multi-institutional Grid environment.

Authorization services are important to Grids. There must be mechanisms in place that will determine what Grid users can and cannot access, as well as to what degree they can access resources (read, write, etc). The proposed brokered approach manages all access control assertions (policies) on the Common Grid Layer. All access control is determined on a higher level of interaction than purely site-to-site assertions.

The broker maintains a directory structure of all resources and users participating in the Grid. This directory structure stores all user access credentials to a resource. As highlighted in previous sections (section 6.2.2) resources have users and owners. Owners can specify access to the resource. Owners set the access control assertions to their resources. A resource owner will make a resource available to another organization or user base. When a user wishes to use a resource, the broker will test their authenticated credentials against their access rights to a resource. The broker will only grant access to that user if they have sufficient access to access the resource they wish to use.

6.3.3 Execution Management

Execution management is a key capability of the Open Grid Service Architecture (OGSA) (Foster, et al, 2006). Execution Management Services (OGSA-EMS) are concerned with the problems of instantiating and managing, to completion, units of work (Ibid). Execution management was discussed as an OGSA capability in section 3.2.2. In order for a brokered approach to be implemented within an OGSA context, the broker must consider how execution management would be affected if authentication and authorization are abstracted to a higher level.

For the purpose of this framework, execution management must be abstracted to the broker. The broker will be able to match a user request with resources. The proposed approach will allow the broker to have knowledge of the user's access rights and resources available on the Grid. This allows the broker to effectively decide what resources a user is assigned when a request is made.

6.3.4 Scheduling Services

Scheduling services are linked to execution management services. The Grid job scheduler will be abstracted to the broker as well. This is in order to maintain consistency in the execution of jobs, and to ensure resources are not overloaded with work. This will allow broker to schedule jobs on the appropriate resources for a user request. The broker will be able to determine the type of resource needed to complete the requestor's computation, i.e., processor requirements, storage requirements, etc.

6.3.5 Network Communications

Network communications are not explicitly defined as a requirement of the broker, but are implied due to the nature of the broker. All the communications that occur between the broker, users and resources are typically over some form of network. There are a wide range of network considerations when implementing a Grid broker. These considerations include: difference in link speed (or network latency), different protocols used by various Grid participants, and secure and reliable information exchange.

Typically TCP\IP protocol is used to share information between Grid participants over a public network, such as the Internet. There are standard security technologies, which can be utilized to secure communications between parties, such as SSL, etc. However, a range of other protocols may be utilised. When implementing a broker these network requirements must be considered.

6.3.7 Storage (Metadata and Information)

The broker will require some storage area to store the directory and other information required for the Grid to function.

6.4 Evaluation Of A Brokered Approach To STRIDE

The following table maps the identified broker services to STRIDE. This section will discuss how each Grid service is at risk from the relevant STRIDE threat category (Depicted in table).

Table 6.1: Applicable STIDE threat categories to broker services

	Spoofing	**Tampering**	**Repudiation**	**Information Disclosure**	**Denial of Service**	**Escalation of privileges**
Authentication	X					
Authorization	X	X	X		X	X
Execution management					X	
Scheduling Services				X	X	
Network communications	X	X	X	X	X	
Storage (Metadata and information)		X	X	X	X	

6.4.1 Authentication

OGSA Grid security is handled through WS-Security (WS-S) specification and Grid Security Infrastructure (GSI). X.509 Public Key Infrastructure is extended to support Grid authentication (Li, Cui, Tian, 2006).

Grid participants authenticate to the Grid via network connections. Grids typically communicate over public networks, such as the Internet. Network authentication services are primarily at risk from spoofing attacks, especially over public networks. A hacker will make use of "man in the middle" attacks to intercept the network communication between the Grid participant and the broker (see

section 7.1.1.2). The hacker can obtain the users authentication credentials to gain access to the Grid.

A number of standard security services can be employed to minimise the risk of hackers obtaining user credentials via "man in the middle" attacks. One such service is encryption. The contents of packets between communicating parties can be encrypted utilizing some method of Public\Private key encryption (PKI). The communication channel can be secured utilizing Secure Sockets Layer (SSL) or any other method of communication encryption.

6.4.2 Authorization

Within the proposed broker approach, authorization requests are make to the broker, and the broker will determine a users rights, based on the access policy defined by the resource owner. The directory structure storing resource information and access policy information is sorted on the broker site. Authorization assertions in current Grid implementations are handled utilizing a X.509 based PKI infrastructure (Welch, et al, 2003).

Authorization services within the proposed broker context are vulnerable to spoofing attacks, denial-of-service (DoS) attacks, and tampering. Escalation of privileges and repudiation are results of tampering.

Authorization services are at risk of spoofing attacks in the same way authentication services are. Authorization information is shared between Grid participants over a network. This service is at high risk of "man in the middle" attacks. A common method to combat these attacks is the use of PKI. PKI allows for the protection of information, preserving its confidentiality, as well as protecting its integrity, by insuring the information was received from the correct party.

An attacker can employ several methods to prevent the desired network packets with vital authorization information to not reach its destination. This attack method will "stall" the process between service requestor and service provider, possibly resulting in time outs, ultimately, preventing the service requestor to gain

access to the desired service. This form of attack is often a last ditch attempt by a hacker, if they cannot compromise the desired target.

User access policies are stored on a directory within the broker. This directory is at a risk of being compromised by attackers. An attacker could gain access to the directory by compromising the server the service is hosted on. If this directory were to be compromised by an attacker, and the attacker alters the information within it. This could result in the escalation of user privileges, if a low access account is promoted to super-user status. The attacker could then delete the transaction logs for the service to hide his\her tracks, and this could result in repudiation. A typical method to secure this type of service is through a process of machine hardening (close un-used ports, firewalls, update system software, etc).

6.4.3 Execution Management

Execution management is a service hosted within the broker. This is in order to allow execution services to have access to authentication and authorization services, and to be accessed by scheduling services. This allows for the Grid to provide a more effective service to its users. Execution management shares similar vulnerabilities to the authorization service, due to the similarities in the services in terms of implementation. Execution management services are primarily vulnerable to denial-of-service attacks. Attackers will employ network or host denial of service methods against execution management services.

6.4.4 Scheduling Services

Scheduling services are integrated into execution management services. Scheduling services share the similar security concerns as execution management, due the nature of the service and its similarity in implementation to execution management. Scheduling services primary security concerns when compared to STRIDE include information disclosure, and denial-of-service (DoS).

If the underlying server hosting the Grid scheduler is compromised, an attacker will have access to the Grid scheduler information. The scheduler handles and exchanges vast amounts of data. An attacker could compromise the scheduler

in order to obtain corporate secretes. In this event, the Grid scheduler becomes compromised or crippled in anyway. This could result in the Grid being unable to schedule user requests, ultimately resulting in denial-of-service. Machine hardening can be used to combat the compromise of the machine scheduler.

6.4.5 Network Communication

Network communications services are not an explicit component of the broker service. However, network communication services facilitate and support many of the broker's services therefore, it must be considered. Network communications are vulnerable to spoofing, tampering, repudiation, information disclosure and denial-of-service attacks. Typical methods employed by OGSA and Globus to secure network communications include encryption of the data exchanged between participants (X.509 PKI infrastructure); as well as the encryption of network communication pipes (VPNs, etc).

6.4.6 Storage

Storage services are not an explicit component of the broker service. However, like network communication services, storage services provide vital services to the broker. If storage services are compromised, the normal operation of the broker will be affected. Therefore, storage must be considered when evaluating the proposed broker's security infrastructure. When evaluating storage services against STRIDE, it is vulnerable to the following threats tampering, repudiation, information disclosure, and denial-of-service. Standard measures to protect broker data include regular backups of data, putting in place redundant hardware (RAID, etc), or site replication (data replication).

6.5 Conclusion

Security issues affecting the Common Grid Layer were identified. A brokered approach to providing Grid-wide authentication and authorization was discussed. This approach addresses the security needs of the Grid on the Common Grid Layer, which primary include authentication and authorization of Grid participants when accessing external sites to their own. Addressing authentication and authorization issues on the Common Grid Layer were found to address other key issues; such as single sign-on, delegation, and the forming of complex user-based trust relationships. The STRIDE model was discussed. Microsoft‘s STRIDE threat classification scheme was used to classify threats Grid services (as Web-based applications). Services required by the proposed broker were identified and discussed, and then compared against STRIDE for security vulnerabilities.

The next chapter will discuss a security strategy for securing Grids at the Local Grid Layer.

Chapter 7

Grid Resource Threat Modelling Methodology

Chapter 6 discussed a Grid broker that can be utilized on the Common Grid Layer. The purpose of the broker is to facilitate high-level trust relationships between Grid participants. This chapter is concerned with the security of a Grid resource on the Local Grid Layer. The goal of this chapter is to discuss a generic threat-modelling technique. This threat-modelling technique can be used by Grid designers to secure Grid resources.

To understand the importance of securing one's Grid, it is important to understand the threats and impacts associated with insufficient security (Whitman, 2003). Grids facilitate large-scale collaboration between globally dispersed parties with varying levels of trust between one another. Grids primarily operate over public network infrastructure, such as the Internet (Foster, 2000). Due to the open nature of the Internet, Grids are at a greater risk of being attacked compared to closed systems (systems behind a corporate firewall) (Surridge, Upstill, 2003). Attackers might employ a variety of methods to attack a Grid, but their actions can be grouped into a set of generic actions. STRIDE will be used in this chapter to categorize (group) attacker's goals when attacking a Grid. STRIDE was discussed in chapter 5, section 5.2.1.

The Local Resource Layer is concerned with the security challenges faced by a single Grid site. The single site could host one or many Grid resources. This layer is faced with more traditional information security challenges, such as operating system and network security. Grids are heterogeneous in nature and are implemented on a wide variety of hardware and software.

Whitman identifies two components to a successful information security strategy, which can be applied to Grids. Firstly, one must know what the threats faced by a Grid are, and, secondly one must know the vulnerabilities of a Grid (Whitman, 2003).

This chapter will discuss the anatomy of an attack, that is, the generic methodology used by attackers when performing an attack. Knowing how an attacker performs his\her attacks against a Grid will better prepare security administrators when defining a Grid security strategy. A Grid resource threat-modeling technique will be discussed. This technique will follow the generic steps of identifying threats to a Grid resource. STRIDE will be used as a basis for identifying threats to Grid resources. Once Grid threats have been identified, applying countermeasures to protect these resources will be discussed.

To summarize, good information security strategy begins with knowing what one's weaknesses or potential attack points are, as well the possible external threat agents that threaten one's system (Grid) (Whitman, 2003). The goals of this chapter are to:

- Identify a generic attack methodology (*knowing one's external threats*),
- Identifying a threat-modelling process Grid designers can use to identify their Grid assets and potential weaknesses (*knowing one's weaknesses*), and
- Discuss countermeasures that can be applied to Grid resources to minimize risk of attack.

7.1 Threats and Threat Modelling

Understanding the basic approach used by an attacker will better equip Grid designers to understand how their Grids are at risk from attacks, and how to best secure them. By thinking like attackers and being aware of their likely actions, one can be more effective when applying countermeasures to protect Grid systems (Meier, et al, 2003). This section will discuss a five-stage process generally employed by attackers when attacking an online Web-based system, such as a Grid.

Microsoft (Ibid) identifies the basic five step attack approach generally used by attackers. This approach defines generic steps an attacker will need to perform in order to complete a successful attack. Not all steps are required in every instance. The steps in Meier et al's attacker's methodology are listed below, and figure 7.1 shows a graphical representation of this process (Ibid):

- **Survey and assess** - This is the initial stage of the hacking process. The hacker will try to learn of possible servers and services on the network. The hacker will then try to find possible weaknesses and exploits them to try and gain access to the target machine.
- **Exploit and penetrate** - Once the hacker completes the survey phase, the next step is to exploit and penetrate the target. The hacker will try to gain entry to the targeted Grid by exploiting a vulnerability discovered in the survey step of the process. Once the attacker gains entry, he\she will attempt to drop the attack payload.
- **Escalate privileges** - Upon completing the attack and delivering the payload, the hacker will then attempt to gain administrative access to the Grid.
- **Maintain access** - If the attacker successfully gains administrative privileges, he\she will try maintaining access to the compromised Grid. This will make future access easier.
- **Deny service** - If the attacker is not successful in his\her attack, he\she will try launching a Denial of Service attack (DoS) against the targeted Grid. The purpose of this is to deny legitimate use of the service.

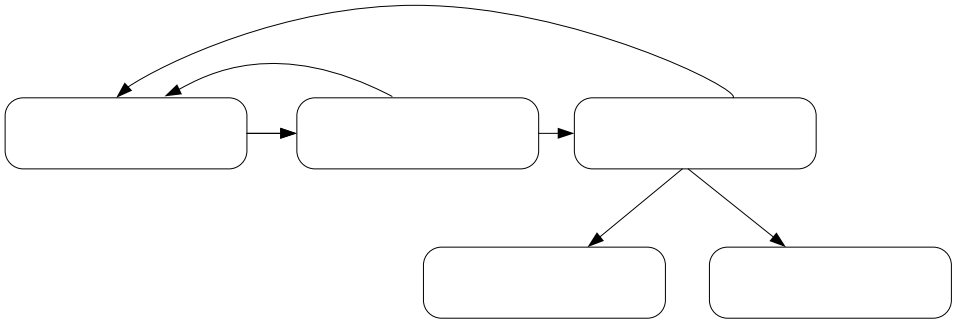

Figure 7.1: Steps in a typical attack (Meier, et al, 2003, pg 15)

7.1.1 Survey and Assess

This is the initial stage to an attack. The attacker will try and identify servers and nodes on the targeted network. Typically this is achieved by using packet sniffers[1] and ping sweep tools[2]. Once potential targets are identified, the attacker will use a variety of tools to learn more about the target. Typical methods of learning about a target include port scanners and banner grabbers, among others.

A port scan will show the attacker what ports are open, and what services are running on the target machine. Banner grabbers will show versions and vendor information of the services running on the machine. This is all potentially useful information to the attacker when determining if the machine is a good target.

The attacker will then pick his\her target based on the information that has been gathered. The attacker has some criteria for the selection of a target. This is typically the attacker's knowledge of weakness in the Grids software (vendor implementations of services and versions), or if he\she is pursuing a particular goal, i.e. theft of information, Denial-of-Service, information disclosure, etc.

7.1.2 Exploit and Penetrate

Once the attacker has identified a target, the next step is an attempt to exploit and penetrate the target. The attacker will exploit the targeted Grid via a vulnerability that he\she has identified in the 'survey and assess' stage. A variety of attacks can be employed by the hacker at this stage to compromise a Grid. Some notable attacks include:

- **Brute force:** This method is used to guess passwords. Brute force attacks generate a list of every possible keystroke combination that can be entered by a user, and then passes them one by one to the targeted Grid authentication mechanism until a match is found.

[1] Packet sniffers are common tools used by attackers to intercept TCP/IP communications between communicating parties. Ethereal is a commonly used packet sniffing utility (ethereal, 2006).

[2] A ping sweep (also known as an ICMP sweep) is a basic network scanning technique used to determine if a range of IP addresses map to live hosts (computers).

- **Dictionary attack:** This method works in a similar fashion to a brute force attack, but uses a set of predefined words to guess user credentials on a target machine. These lists of words are typically stored in a text file; they are typically digital versions of thorough commercial dictionaries.
- **Buffer overflow attacks:** This method exploits bad coding in a Grid to execute malicious code on a target machine. A buffer overflow occurs when more data is passed to a program then it has made provision for in memory. This results in the code crashing and the undesired malicious code executing.
- **SQL injection:** SQL injections exploit vulnerabilities in input validations of a Database Management System (DBMS). The result of SQL injection allows an attacker to run arbitrary code on the targeted remote database server. If applications do not validate their SQL queries to the DMBS, an attacker could insert undesired SQL statements within legitimate statements to execute undesired commands on the targeted machine. Underlying operating system commands could be executed as well as DBMS commands.
- **Cookie relay attacks:** An attacker could use network monitoring software, such as a packet sniffer, to capture a legitimate user's authentication cookie. Once the cookie is obtained the attacker will relay it back to the server and obtain access illegitimately.
- **Man in the middle attacks:** The attacker 'sits' between legitimately communicating parties and utilizes network monitoring software, such as a packet sniffer, to intercept messages. The attacker will either save relevant information, such as authentication credentials, or alter the information to gain access to a Grid.

A wide range of attacks and further variations of these attacks can be used by attackers to exploit and penetrate Grid systems.

7.1.3 Escalate Privileges

Once the attacker has compromised the targeted Grid, the next stage is to try obtaining higher-level access. This is typically administrative-level access. The attacker will attempt to create an administrative-level account, or try to promote a compromised account, if it is not a higher-level access account already.

Depending on the success of attempting to gain higher level access, the attacker will perform one of two actions. Either he\she will try maintaining access to the compromised Grid, or launching a Denial-of-Service (DoS) attack against it. Denial-of-Service attacks deny legitimate users access to the Grid (see section 7.1.5).

7.1.4 Maintain Access

If the attacker successfully obtains administrative access to the compromised Grid, the attacker will try to maintain access to that Grid. This allows the attacker to make future access to the Grid easier, and will make the process of clearing his\her tracks easier. Typically, the attacker will clear his\her tracks by deleting log entries. This results in non-repudiation. Non-repudiation prevents Grid security administrators knowing who performed what action.

The attacker will, typically, plant a back-door application to maintain access to the compromised Grid. A back-door application is defined as “a hardware or software-based hidden entrance to a computer system that can be used to bypass the system’s security policies” (Microsoft, 2006).

7.1.5 Deny Service

If the attacker cannot successfully launch the desired attack against the targeted Grid, the last course of action would, typically, be to try denying legitimate users access to it. This is achieved through Denial-of-Service attacks. The most common method of these attacks is attacking the network connection between the service and its users. Internet-based applications, such as Grids, are at highly vulnerable to Denial-of-Service attacks (Houle, Weaver, 2001).

7.2 Threat Modelling Methodology

"Knowing your systems weakness as well as the possible threats to your system is a first step to developing an appropriate security strategy" (Whitman, 2003). In chapter 5 STRIDE was introduced as a threat-classification scheme. A generic attack methodology often employed by hackers when attacking Grids was discussed in the previous section. The goal of this section is to discuss a threat-modelling process Grid designers and security administrators can apply to the Local Grid Layer.

Threat modelling should be a constant in a Grid's lifetime, and not just a consideration during the design phase of the System Development Life Cycle (SDLC). The two main reasons for this are (Meier, et al, 2003):

1. It is impossible to identify all possible threats faced by a Grid in one go.
2. Grids are rarely static and often change to meet their changing business\users requirements.

7.2.1 Threat Modelling Principles

The following six step threat modelling process can be applied to almost any application security strategy, but has particular relevance to Web-based applications and Grid systems. Consider the following steps discussed by Meier for securing Web-based information systems, such as Grids. (Meier, et al, 2003):

1. **Identify assets:** Identify the assets associated with the Grid that must be protected. Grid assets could include databases or a specialized piece of hardware.
2. **Create an architecture overview:** The architecture of the local Grid resource can be modelled using diagrams and tables. The following aspects of the resource should be identified: the services it is realised on; what sub-systems it consists of; and what the trust boundaries that exist between the local Grid and its external Grid users are.
3. **Decompose the Grid:** The purpose of decomposing the Grid is to identify the autonomous components of the Grid. These components include:

underlying hosts, networks, OGSA compatible components, etc. This exercise is the first step in identifying a local Grid resource security strategy. Profiling the components of the local Grid resource individually can help identify possible vulnerabilities to the Grid.

4. **Identify threats:** During this step, Grid designers identify possible threats that could affect the local Grid resource. This step will require multiple parties within the Grid design and administration groups to co-operate with each other. An understanding of the composition of the Grid is required (Step 3), along with a working knowledge of STRIDE (in terms of relevant attack categories, see chapter 5). With this knowledge at hand and the relevant expertise available, team members can brainstorm and identify threats to the Grid resource effectively (Meier, et al, 2003).
5. **Document the threats:** During this step, the threats identified in step 4 are documented, utilizing a common set template that defines a core set of attributes to be captured for each threat. It is good practice to utilize a common template as this practice establishes a common framework for all parties involved. This will allow for better communication of the threats.
6. **Rate the threats:** Once the threats have been formally documented, key members of the Grid implementation process should rate them. Threats should be rated and prioritized from most significant to least. A simple weighting process can be used. The most common criterion for rating threats is “the probability of the threat against the likelihood of attack”.

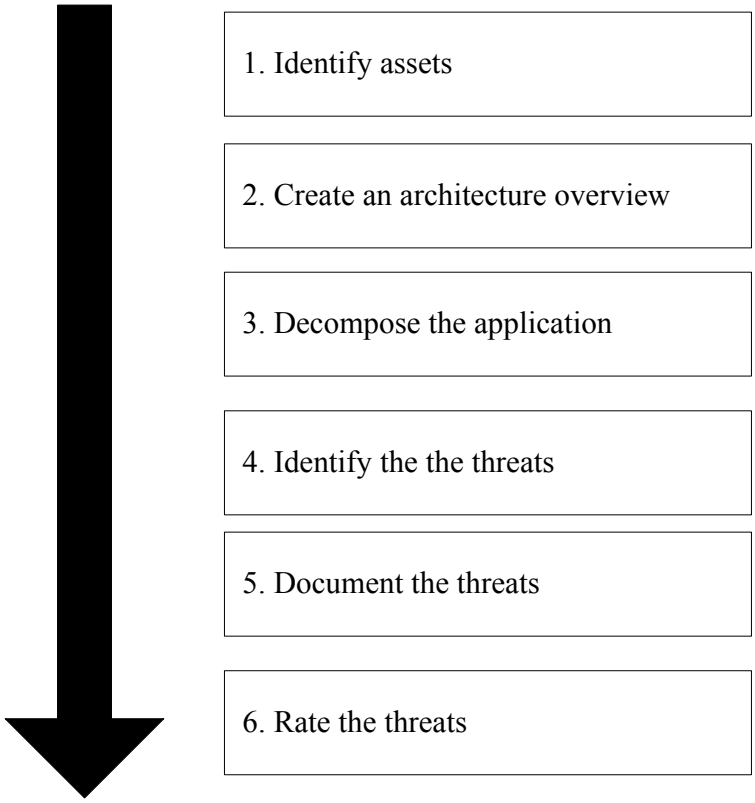

Figure 7.2: Six step threat modelling process (Meier, et al, 2003, pg 47)

Figure 7.2 depicts a graphical representation of the process. Successful completion of a step requires all its prerequisite steps to be completed. The output of this process should be an easily understood document or set of documents. The local Grid resource administrative team should use these documents as a common vocabulary. These documents identify what the threats are that need to be addressed, and how they will be addressed. The following sections will discuss the steps in the six-step threat-modelling process in more detail, when it is applied to a Grid.

7.2.1.1 Identify Assets

Firstly, all assets that make up the Grid resource need to be identified. This could include databases, hardware assets, storage resource, computational assets, etc. Identifying these assets will provide the local Grid resource security administrators with a clear picture of what the valuable components of the Grid resource are, as well as those that would have a financial figure associated with them if compromised, lost, or if contents were divulged to unwanted third parties.

Grids are multilayered structures. Each layer must be considered when identifying the assets that make up a single Grid resource. These layers include:

- The hosting environment,
- The underlying operating system,
- The Grid fabric,
- The OGSA middleware and Web-services, and
- Grid applications.

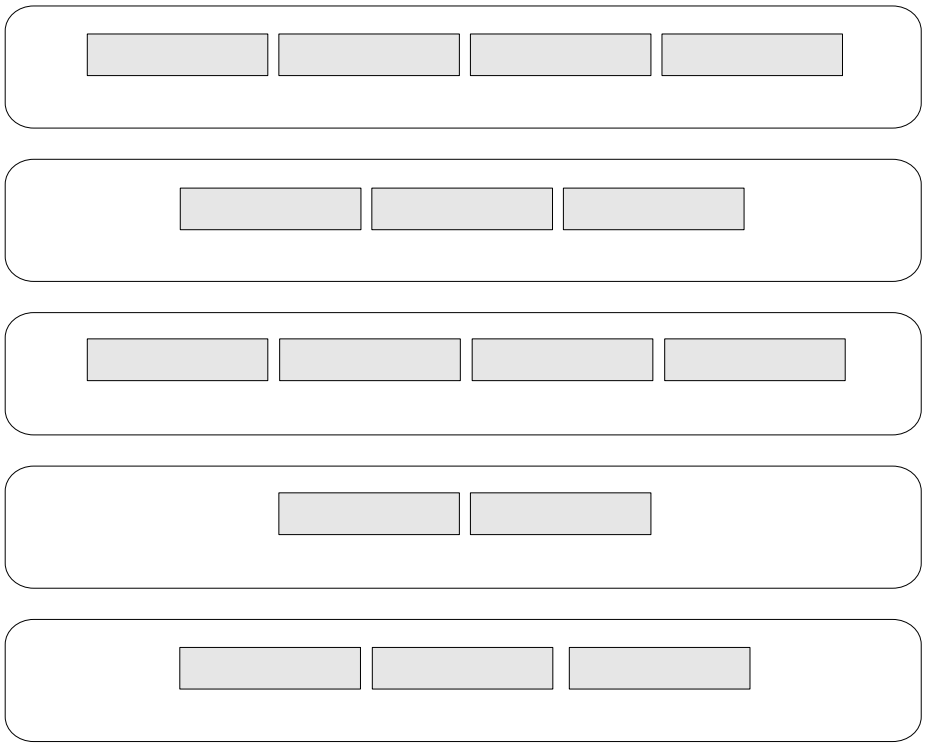

Figure 7.3: Layers of a Grid Resource

The *hosting environment* is primarily responsible for supporting the Grid service. OGSA defines the semantics of a Grid resource instance: how it is created, how it is named, how its lifetime is determined, how to communicate with it, etc (Foster, et al, 2002). However OGSA does not define what the Grid resource does, or how it performs its intended operations. OGSA does not address issues of implementing a programming model, programming language, implementation tools or execution environment (Ibid). This is the role of the hosting environment. Web-services are the driving technology for OGSA Grid functionality. Web-services are easily implemented and maintained in container or component-based hosting environments, such as J2EE, Web-sphere, SUN ONE, or .NET framework (Ibid). These environments abstract the complexities of implementing Web-service components.

The primary purpose of the *underlying operating system* is to support the chosen hosting environment in the same fashion that the hosting environment supports OGSA services. Hosting environments require an underlying platform to support their functions. Hosting environments provide container functionality, but cannot manage hardware, or interact with system resources directly.

The *Grid fabric* is simply any native service or component that is exposed to the Grid via OGSA middleware. The Grid fabric could consist of specialized hardware, or any form of software, exposed to the Grid via OGSA interfaces, to provide computing or storage resources to Grid users. OGSA manages the publishing and discovery of these services.

OGSA middleware and Web-services address heterogeneity in distributed Grid systems. OGSA based middleware, such as the Globus Toolkit (See chapter 3, section 3.3), provide uniformity through a standard set of interfaces to the underlying resources (Globe, De Roure, 2002).

Grid applications are the applications that operate within the Grid in a virtual organization (VO) context. Grid applications are the end result of all the lower-level layers and aspects of Grid architecture working together.

When identifying all Grid resource assets, the above mentioned layers must be considered. A methodical approach to identify assets utilizing the layers of a

Grid resource discussed is recommended. It is recommended Grid resource managers utilize the layers discussed in this section as part of a methodical top-to-bottom approach to identifying Grid assets.

7.2.1.2 Create an architecture overview

The goal of this step is to document the Grid resources assets. An architecture diagram depicting the local Grid architecture and the components that makes up the Grid resource (OS components, DBMSs, Globus middleware and tools, OGSA compatible components,
etc), as well as how these components interact with each other to provide a service, are the primary output of this step.

Identifying how the Grid resource is meant to be used could provide some insight into how it is not meant to be used (Meier, et al. 2003). During this step, Grid resource designers and security administrators should identify how the Grid-resource users access the assets identified in step 1. This will provide greater understanding on how the assets could be misused.

A high-level Grid architecture diagram should be drawn up. The diagram should highlight the components and structure of the Grid resource. Depending on the complexity of the Grid resource and its implementation, multiple architecture documents might be drawn up. Each of these documents could focus on specific areas within the overall picture. Figure 7.4 shows an example of a high-level Grid resource architecture diagram.

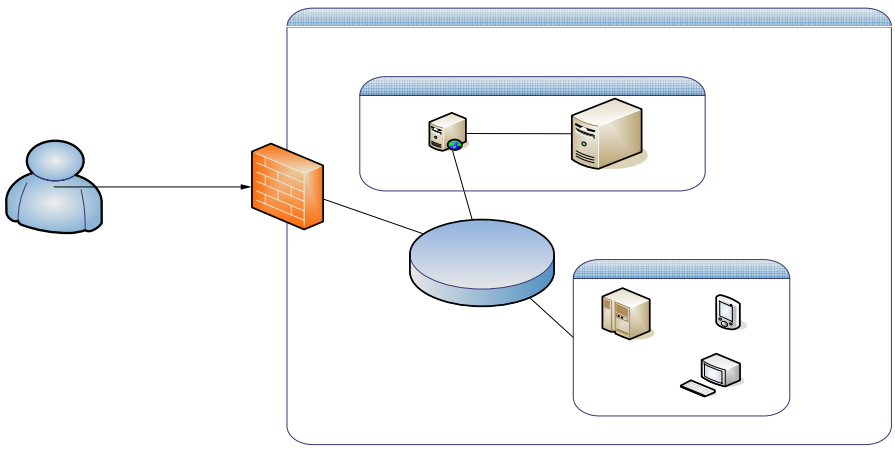

Figure 7.4: Simple example of a Grid architecture diagram

Technologies used in the Grid resource implementation should be identified on the architecture diagrams. The process of listing and identifying the technologies and platforms utilized in the composition of the Grid resource will allow for the identification of technology-specific threats. This will possibly help identify the best mitigation techniques, patches, updates, etc. A list of technologies could simply be listed in a table, as shown in table 7.1 below:

Table 7.1: Example of a list of Grid technologies

Component	Grid resource layer	Implementation details
J2EE	Hosing environment	Underlying runtime environment.
Globus toolkit, version 4	OGSA middleware	Implementation of OGSA compatible interfaces

The goal of this step is to:

- Identify what service(s) the Grid resource expose, or make available to the VO.
- Diagrams and documentation of the local Grid architecture.
- Identify the technologies utilized in the implementation of the Grid resource.

7.2.1.3 Decompose the application

This step requires Grid resource managers to logically breakdown the Grid resource. This is for the purpose of defining a security profile. Breaking down the Grid into its aggregate components will help the Grid resource managers identify the vulnerabilities faced by individual components. Important considerations when decomposing the Grid resource are: identifying trust boundaries, dataflow, entry points, and identifying segments of privileged code.

Identifying the Grid resource trust boundaries requires the relationship between components, assets, and users to be explicitly clarified. The path to assets must be identified from the user interaction perspective (what mechanisms are in

place to protect assets) as well as how a user's request is handled from the first component in the interaction, to the lowest-layer protected component\asset.

The identification of the dataflow can be done concurrently with the identification of trust boundaries. Dataflow is the path taken by data from origination to destination that includes all nodes through which the data travels. The simplest way to identify the flow of data within a Grid resource is to start at the highest level of user interaction. This is, typically, the proxy between the local Grid resource and Grid broker. Dataflow diagrams (DFD) can be drawn up to show the flow of data in the Grid resource. Dataflow diagrams are Unified Modelling Language (UML) diagrams, typically used to depict the interaction of components in a system. It is out of the scope of this research to explain how dataflow diagrams are drawn up. However there are many good UML resources available (refer to "System analysis and design" by Kendall and Kendall (2002)). Figure 7.5 depicts a high-level example of a typical Grid resource DFD.

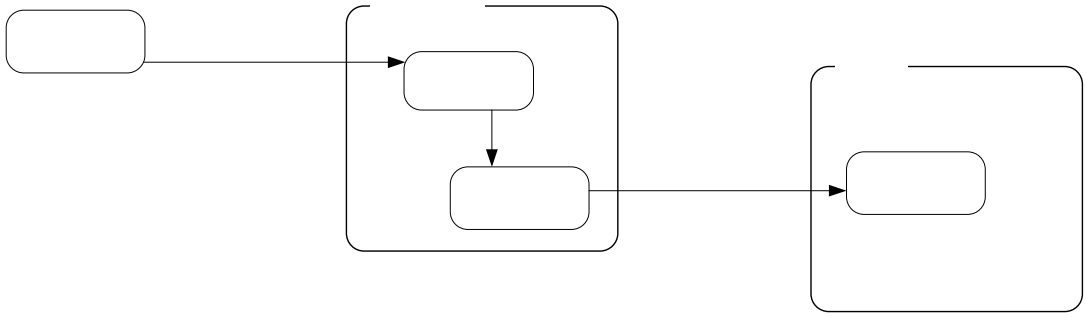

Figure 7.5: an example of a simple DFD

Entry points to the Grid resource are potential points of entry for attackers. Appropriate "gatekeepers" should be identified for entry points (Meier, et al, 2003). Firewalls implementing well-defined access-control policies are the most widely implemented "gatekeepers".

The privileged code segments are the restricted functionality area of the Grid fabric. These code segments perform computations for service requestors, but are never accessed directly by the requestor. Management services defined in the middle layer (service layer) of OGSA are also considered privileged code segments (Foster, et al, 2006). During this step, the following must be performed:

- Identify trust boundaries within the Grid resource.
- Identify the dataflow from when a request comes via the common layer broker.
- Identify all possible entry points to the Grid resource.
- Identify the privileged code segments, both in the Grid fabric and the OGSA middle layer.

7.2.1.4 Identify the threats

Based on all the information collected collectively from previous steps in the threat modelling process combined with working knowledge of STRIDE, one can begin to identify threats that face the local Grid resource(s).

The STRIDE model to classify threats was discussed in chapter 5. STRIDE allows Grid designers to have a checklist of possible attacker goals. STRIDE defines six threat categories (Meier, et al, 2003). Each threat category can be used as a checklist of attacker goals or methods when attacking a Grid (Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service, and Elevation of Privileges). Assessing vulnerabilities to Grid assets based on the simple criteria of what the attacker's intentions and goals are, is a less time consuming and cheaper process than identifying all known vulnerabilities and available exploits for the underlying technologies the Grid resource is built on.

Grid assets need to be analyzed for susceptibility to the STRIDE threat categories. In step 1 (Identify assets), the assets that make up the Grid were identified. Five layers of a Grid resource were discussed (the hosting environment, the underlying operating system, the Grid fabric, the OGSA middleware and Web-services, and Grid applications). The roles of these layers were discussed. This step

required the identification of threats to assets belonging to all layers of a Grid resource.

STRIDE was used to abstract an attacker's intentions into six attack categories. The properties of a Grid resource can also be abstracted in a similar fashion. Grid resource layers can have primarily one of three risk profiles, but may be vulnerable to threats from another threat profile. These profiles include:

- Network threats
- Host threats
- Application threats

Network threats are direct threats to the Grids network infrastructure. Grid network threats are also concerned with the attacks that are carried out against a Grid, using the network communication links between participating sites as the primarily attack vehicle.

Host threats are directed against the system software the Grid is hosted on\built on. This includes all operating systems, hosting environments, Database Management Systems (DBMSs).

Application threats are the threats faced by Grid applications. Grid applications are the high-level applications built on the lower-level building blocks described in this section.

Table 7.2 provides a checklist of all previously identified Grid resource layers. Each layer has one ore more threat profiles, and these profiles are ticked in the appropriate column. It is interesting to note all layers are vulnerable to host-related threats.

Table 7.2: Table of Grid resource layers and Grid threat profiles

Grid resource layer	Network threats	Host threats	Application threats
Hosting environment		X	X
Underlying operating system		X	
Grid fabric	X	X	
OGSA middleware		X	X
Grid applications	X	X	X

When identifying threats to the Grid, consideration must be made to the following general Grid-resource threat profiles:

- Network threats
- Host threats
- Grid application threats

7.2.1.5 Document the threats

When documenting threats faced by the local Grid resource, standard templates must be used. Templates must contain attribute information about the threat, and any assumptions made. Utilizing standard templates makes communicating threats to all members of the Grid resource managers easier. It provides a common framework of understanding.

7.2.1.6 Rate the threats

At this stage there should be lists of possible threats faced by each aggregate Grid resource component. The next step is to rate these threats in order of danger they pose to the Grid resource. This process is known as risk assessment (Whitman, Mattord, 2003, pg140). It might not be financially or logistically viable to address all the possible threats identified (Meier, et al, 2003). Some threats might be ignored as the possibility of their occurring and the potential payload would be minimal.

A common weighting scheme can be used to rate a threat. Based on the threats score and danger it poses, it can be dealt with accordingly (Meier, et al, 2003; Whitman, Mattord, 2003). The risk of a threat can be expressed as **Risk = Probability * Damage Potential.**

The *probability* factor in the above calculation is the likelihood a particular vulnerability could be exploited by attackers. Grid resource managers should provide a rating of the possibility of a threat occurring. The rating can be expressed as a numeric value from 1-10. A value of '1' indicates the threat is not very likely to be exploited, and '10' indicates the threat will almost certainly be exploited.

The *damage potential* factor depicts a numeric representation of how valuable the asset is to the Grid. Damage potential can be expressed as a numeric value from 1-10, similarly to probability. A value of '1' would indicate if the asset was to be compromised, the Grid would not notice the effects or the value of replacing it are minimal. However, a rating of '10' indicates if the asset was to be compromised or lost, the reputations would be considerable and noticeable.

The two values are multiplied in the risk calculation. Once this is done for all identified threats, the prominent threats will be identified and ranked. A risk rating of '100' indicates the risk to be clear and present. If it were to be exploited, it would result in noticeable losses. A risk rating of '1' indicates minimal importance of the risk. A list of the risks can be made from order of significant to least significant, i.e., from 100 to 1. The process of identifying threats is an iterative process. It must be performed periodically (Meier, et al, 2003, pg 65; Whitman, Mattord, 2003; Surridge, Upstill, 2003). Once the risks have been rated from significant to least significant, a process of identifying controls to mitigate the effects needs to be undertaken.

7.3 Applying Countermeasures To Grid Threats

In the previous section, a six-step threat-modelling technique was discussed (Meier, et al, 2003; Whitman & Mattord, 2003). This threat-modelling technique was adapted to identify threats faced by Grids. It was discovered the primary output of the threat-modelling exercise is to identify the most pertinent threats to the Grid resource. This output is achieved by identifying the threats to the Grid resource; and then rating these threats. Threats are rated according to probability of their occurring and the damage potential of a threat, if it was to be exploited. Once all the pertinent threats have been rated in order of most impactful, to least impactful, Grid resource managers must assign effective countermeasures to these threats.

These threats were identified from the perspective of an attacker with the STRIDE threat categories as possible outcomes. STRIDE can be applied as a method to identify countermeasures as well. Each threat category described by STRIDE has a corresponding set of countermeasure techniques that should be used to reduce risk (Meier, at al, 2003). Meier defines a general table of countermeasures for each STRIDE category. The following table has been adapted specifically for the Grid environment:

Table 7.3: A list of countermeasures for STRIDE threats to a Grid resource (Meier, et al, 2003, pg 17-18)

Threat	Countermeasure
Spoofing (user identity)	• Use strong authentication • Encrypt or hash secrets (passwords, etc) • Encrypt credentials when they are sent over a network connection • Protect authentication tickets with Secure Sockets Layer (SSL)
Tampering (with data)	• Use data hashing • Use digital signatures • Use strong authentication • Use tamper resistant protocols over

	communication links • Secure communication links with protocols that provide message integrity
Repudiation	• Use secure audit trails • Use digital signatures
Information Disclosure	• Use strong authorization • Use strong encryption • Secure communication links with protocols that provide message confidentiality • Do not store secrets in plain text
Denial-of-Service	• Use bandwidth throttling techniques • Validate and filter inputs
Elevation of Privileges	• Follow the principles of least privileged use • Use lest privileged credentials to run services

The following section will discuss a few countermeasures relevant to each STRIDE threat category. This section will provide Grid resource managers with some background and understanding of what countermeasures are currently available and in use.

7.3.1 Security Services

7.3.1.1 Strong authentication

X.509 public key certificates are commonly used in Grid implementations for providing authentication between parties. The Globus toolkit (discussed in chapter 3, section 3.3) implements X.509 security services in the Grid Security Infrastructure (GSI) portion of the toolkit (Welch, Foster, Kesselman, Mulmo, Pearlman, Tuecke, Gawor, Meder, Siebenlist, 2004).

When utilizing X.509 certificates as an authentication mechanism between distributed parties within a Grid, a certification authority (CA) can be used to issue certificates to Grid users. A certification authority is not always required, virtual

organizations can make use of their own certification service. Certification authorities are trusted third parties that issue Internet users (or Grid users) with digital certificates to verify who they are to other Internet (Grid) users. Certification authorities rely on a complex hierarchy of trust relationships between each other to maintain a state of universal trust (Schneier, 2000, pg 232-233).

X.509 certificates provide Grids with the flexibility to allow an entity to trust another organization's certification authority (CA), without requiring that the rest of its organization do so or requiring reciprocation by the trusted CA (Welch, et al, 2004). This feature allows for complex inter-organizational authentication.

7.3.1.2 Hashing

Hashing provides a means to check the integrity of information transmitted over unsecured and un-trusted mediums (Krawczyk, Bellare, Canetti, 1997). Hashing is the process of using a mathematical algorithm against data to produce a numeric value that is representative of that data. Typically a "secret key" is used between parties to hash data transmitted over un-trusted networks. MD5 and SHA-1 are two of the most popular hashing algorithms available. MD5 is the most commonly used hashing algorithm in Grid environments (Humphrey, Thompson, 2005).

7.3.1.3 Encryption

Encryption is a security service that ensures the confidentiality of data. Encryption is the process of transforming "plaintext" into "ciphertext". Ciphertext means text that is hidden but can be restored to the original plaintext by another algorithm (the invocation of which is called *decryption*) (Humphery, Thomposon, 2005). Two common encryption methods are through the use of "symmetric" and "asymmetric" encryption algorithms.

Symmetric algorithms use the same key to encrypt and decrypt a message. Both the sender and receiver in this instance share the same key. In contrast, asymmetric algorithms make use of two keys to encrypt and decrypt messages. Each participant has two keys, a public key and a private key. Due to this, asymmetric encryption is often referred to public-key cryptography. The public key

is made publicly available. This key is used by other users to encrypt messages to the owner of the key. Only the user's private key (which is kept secret) can decrypt the message encrypted with the same user's corresponding public key.

Symmetric and asymmetric encryption algorithms are used in combination to protect Grid information. Typical usage of encryption in Grids involves the data in transit to be encrypted utilizing a symmetric encryption algorithm, such as AES (NIST, 2001). Symmetric encryption algorithms are stronger mathematical algorithms. They are harder to crack (cracking is the process of overcoming protection mechanisms in software or computer systems [Wikipedia, 2006]), and have less system overhead when encrypting and decrypting messages. The symmetric key is then encrypted utilizing an asymmetric algorithm, such as RSA (Rivest, Shamir, Adleman, 1978). The sender's asymmetric private key is used to encrypt the symmetric key, and then the recipient's public key is used to encrypt the message. The encrypted message is then sent to the recipient, the recipient uses the relevant keys to decrypt the message and obtain the symmetric key. This concept is the basis for digital signature implementations.

As discussed in section 7.3.1.1 Grids primarily implement X.509 digital certificates to perform authentication services.

7.3.1.4 Tamper-resistant communication protocols

The most commonly used network communication suit of protocol is TCP/IP. TCP/IP was designed with connectivity in mind, a goal it has achieved. However, TCP/IP is susceptible to tampering attacks. Security was not a concern when the protocol was designed. There are a number of inherent security flaws in the protocol (Bellovin, 1989).

The primary communication protocol used over the Internet is TCP/IP. The Internet is made up of a number of interconnected devices, called routers. Routers determine the path IP packets take from their point of origin (sender) to their destination (receiver). As packets pass from router to router, their contents are open to anyone to read (Schneier, 2000). IP packets not only contain fragments of the data communicated between partied, but sender and recipient information as well.

Attackers can obtain this information and use it for malicious purposes as discussed in previous sections (Spoofing attacks, man-in-the middle attacks, etc).

Even though the TCP/IP has many security flaws, it is still the most widely implemented network communication protocol suite. It might not be feasible to remove TCP/IP support from Grid implementations in favour for another more secure communication protocol. It might be more feasible to use secure communication protocols that can be implemented as an extra layer on top of TCP/IP.

7.3.1.5 Secure communication protocols

Secure communication protocols provide confidentiality and integrity to network communications over insecure networks, such as the Internet. A common public key authentication protocol used in Grids is the Transport Layer Security (TLS) protocol (Dierks, Rescorla, 2004). The Grid Security Infrastructure (GSI) porting of the Globus toolkit is built on top of the TLS protocol (Humphrey, Thompson, 2005). There are many other possible secure protocols that can be used, but this section will focus on TSL as Globus is the most widely implemented Grid middleware.

TSL is derived from the Secure Sockets Layer (SSL) version 3 protocol (Frier, Karlton, Kocher, 1996). TSL makes use of X.509 public key certificates (discussed in section 7.3.1.1) to authenticate the communication requestor.

X.509 digital certificates are used in conjunction with TLS; a X.509 certificate is presented as an authentication token. Once the party is verified (authenticated) by the token, the party is challenged using the TLS handshake protocol to prove its knowledge of the private key associated with the public key in the certificate (Humphery, Thomposon, 2005). After the party has been successfully verified, the communication channel is secured so only the authenticated parties can communicate over it. This ensures confidentiality and integrity over insecure communication links, such as the Internet.

7.3.1.6 Auditing and accounting services

Auditing and accounting services provide non-repudiation to Grid implementations. Non-repudiation prevents a user or attacker from denying he\she had performed an action. Audit records of performed operations provide traceability in the event of a threat or breach (Ramakrishnan, 2004). In order for accounting services to be accurate, Grids must support two requirements each Grid user needs a unique identity across the Grid; and there must be adequate authentication services in place to ensure users are correctly authenticated.

However, there are two challenges when implementing Grid-wide auditing and accounting. These challenges are: the heterogeneity of Grids; that sites might implement incompatible accounting solutions (Windows event logs, UNIX\LINUX Syslog, etc); and accounting information in Grids is dispersed. This makes correlating logs and audit files difficult (Ibid).

A possible solution to these challenges was discussed in chapter 6. A broker could be utilized to provide unique identities to all Grid users, provide authentication services, and single-site coherent user audit and accounting information.

7.3.1.7 Digital signatures

Digital signatures services provide Grids with confidentiality, integrity and authorization. Digital signatures are implemented utilizing public key encryption (discussed in section 7.3.1.3). Grid middleware, such as the Globus toolkit primarily make use of X.509 certificates for digitally signing messages passed between Grid participants (X.509 certificates were discussed in section 7.3.1.1).

The use of digital signatures provides a number of key security services to Grid implementations. Digitally signing communications between Grids at a message level provides end-to-end security between communicating Grid participants (Nagaratnam, et al, 2002). Digital signatures are widely used in Grid-authentication strategies (discussed in section 7.3.1.1).

7.3.1.8 Bandwidth throttling

Bandwidth-throttling security services primarily ensure availability to Grids. Bandwidth-throttling techniques have two purposes in Grid environments. Firsly, they are used to ensure policy-driven Quality-of-Service (QoS). This method works by assigning a priority rating to types of network traffic; less important traffic is throttled: this gives more important traffic preference on the "wire". Another more widely used application of bandwidth throttling is to protect against network-based Denial-of-Service (DoS) attacks. Bandwidth-throttling services are available with most current network infrastructure equipment (routers, switches, network interface cards [NICs]), and are supported in most modern network-enabled operating systems.

7.3.1.9 Input validation

Input-validation services provide Grids with integrity. Input-validation mechanisms protect against buffer overflow, SQL injection, and other input-based attacks. It cannot be implemented utilizing a technology. It has to be build into the Grid service at a code level.

Meier describes some guide lines to be considered when adding input validation at a Grid resource level (Meier, et al, 2003):

- Developers must assume all input is malicious,
- A centralized approach must be utilized,
- Developers must not rely on client-side (requestor) validation, and
- A "constrain, reject, and sanitize" input approach must be adopted.

Meier suggests, "Input validation starts with a fundamental supposition that all input is malicious until proven otherwise" (Meier, et al, 2003). Input from outside the Grids resource trust boundary must be validated.

Utilizing a centralized approach will make implementing input validation easier. For example, input validation could be handled by a single set of libraries. This approach ensures validation is applied consistently across the Grid resource (Meier, et al, 2003).

Input validation must occur on the Grid resource, and not rely on clients (service requestors) to perform input validation. The client validation model assumes input received by the Grid resource is already validated. An attacker could bypass the mechanism on the client side to send malicious invalidated input to the Grid resource.

A preferred approach to performing input validation is to constrain inputs that are allowed by the Grid, at design time of the Grid resource. When the Grid resource is deigned developers should know what input the Grid will expect. It is easier to allow a finite set of known inputs, rather then trying to identify a wider range of illegal inputs. However, for defence in-depth, known malicious inputs can be rejected and then attempts to "sanitize" the input data can be made (Ibid).

7.3.1.10 Least-privileged use model

The least-privileged use model provides Grids with authorization. Least-privileged use is defined as, "A well-known principle in computer security that states that each entity should only have the minimal privilege needed to accomplish its assigned role and no more" (Welch, et al, 2003). Welch reports a least-privileged model is implemented in the Globus toolkit version 3 (GT 3), and newer revisions. When exposing the Grid fabric to Grid users, this model can be used to similar effect.

Public facing services should have minimal or no privileges; this will reduce the impact if they were to be compromised. The attacker will have not advanced in the attack.

7.4. Conclusion

Two essential requirements to a successful security plan were identified. Firstly, Grid resource managers must know the threats their Grid faces, and secondly, the Grid's vulnerabilities must be identified. This chapter identified a generic attack methodology utilized by attackers. This attack methodology provides Grid resource managers with some insight into how an attacker will attack a Grid. A six-step threat-modelling process was proposed. This process was adapted for Grids, and utilized STRIDE to categorize Grid threats and attacker goals when attacking a Grid resource. The threat-modelling process discussed can be utilitized by Grid resource managers when identifying threats to Grid resources on the Local Grid Layer. The threat-modelling process outlined a top-down methodical approach to securing Grid resources on the Local Grid Layer.

Chapter 8

Conclusion

This dissertation proposed a security framework towards holistic Grid security. This security framework proposed two abstract layers of Grid security for consideration. The first layer was concerned with high level political Grid security issues. These issues included authentication, authorization, and the creation and maintenance of complex trust relationships between Grid participants. This layer was identified as the Common Grid Layer.

The second layer was concerned with securing the lower level Grid resources. Grid resources are often build on heterogeneous infrastructure; thus, prescribing a Grid security strategy is particularly challenging. This layer was identified as the Local Resource Layer. The STRIDE threat classification scheme was introduced and discussed. STRIDE is an acronym (Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service, Elevation of privileges) for popular security threats faced by distributed applications, including Grid Services. STRIDE was used extensively in the definition of the holistic Grid framework discussed in this research.

It was suggested that the Grid security challenges on the Common Grid Layer could be solved by utilizing a brokered approach to facilitate trust relationships between Grid participants. The brokered approach utilizes a central abstracted software component in the Grid, which maintains a directory of legitimate Grid users, Grid resources, and the authorization (access control) mappings of users to resources. Implementing a brokered approach addresses some security concerns identified in Chapter 6; e.g. a broker could provide: single-sign on authentication to Grid users, strong and coherent authentication, delegation of user credentials, coherent accounting and auditing, and facilitation of dynamic complex user-based trust relationships. The aspects (components and services) to make a broker possible in an OGSA-based Grid were discussed. These aspects were

individually evaluated against STRIDE for possible vulnerabilities to STRIDE threat categories.

A generic threat modeling process applicable to Grid resources was investigated. This threat modeling process is generic enough to be applicable to the heterogeneous nature of Grid resource implementations, but takes into account Grid specific constraints. The threat modeling chapter identified two general requirements to satisfy a Grid resource security strategy. These requirements are, firstly, threats to the Grid need to be identified, and secondly, the individual Grid services vulnerabilities need to be identified. The generic anatomy of an attack was discussed. It was found that combining a working knowledge of STRIDE, understanding the anatomy of an attack, and having knowledge of the Grid resource assets allowed for a suitable framework to be arrived at.

A six step threat modeling process was adapted to Grids. This threat modeling process provides a means to identify the vulnerabilities to Grid resources. Once threats to the Grid resource were identified, a number of countermeasures were discussed.

Addressing security requirements in each of these two layers of Grid security provided a holistic framework for building secure Grids.

8.1 Revisiting The Problem Statement

This dissertation addressed security concerns in OGSA based Grids. A holistic framework to Grid security was required.

It is contended that a brokered approach to ensuring interoperability in security services in an OGSA Grid context is very feasible. Other broker-mediated functions were also made secure through the security framework suggested. By abstracting a layer of security services beyond the local fabric level, a system of mapping between client systems, the broker system and resource/fabric systems could ensure Grid-wide security interoperability and Grid-Infrastructure level management. The efficacy of this framework depends largely on the broker being aware of all systems in a granular fashion, e.g. what security mechanisms are

employed, what the principals' credentials are and how they are presented in each system, etc.

A comprehensive means for applying threat-modelling in an ongoing fashion to protect Grid resources at the fabric level is also suggested in chapter 7.

8.2 Shortcomings Of The Framework

Grids are still a new and dynamic field of study. There is still a lot of work to be done in all areas of Grid research, particularly Grid security. The OGSA specification is still under review and modification, while the Globus Toolkit is under constant modification by the Globus community.

The brokered approach to Grid security proposed in this dissertation (Chapter 6) can only really serve as a set of recommendations to Grid middleware communities.

8.3 Future Work

Grids research is constantly changing. New standards are being defined, while older ones are under constant review. With that said, the area of discourse should be thoroughly reevaluated in the event of future work on the model proposed in this dissertation. There are two primary areas this research can be taken further.

Firstly, the proposed broker to interoperable Grid security (proposed in Chapter 6) can be incorporated into currently available Grid middleware. The Globus Toolkit is an open source project. The Globus Toolkit could be used as a test basis to implement a basic prototype to practically evaluate the broker.

Secondly, the model could be extended to include Grid specific countermeasures to threats faced by the Local Resource Layer. These countermeasures would have to be identified and researched.

8.4 Final Word

The Grid security field is still a new area of research. With that said, the outcome of this research is not intended to be a final 'set in stone' solution, but rather it is the hopes of the author that it will provide a foundation for future efforts in this area.

A. Appendix A: Published Article

The following article was published in the *Proceedings of the Information Security South Africa (ISSA 2006)*, from Insight to Foresight Conference (ISSA 2006), Sandton, South Africa, July 2006.

A BROKERED APPROACH TO INTEROPERABLE SECURITY IN OGSA-BASED GRID SYSTEMS

Demetrios Loutsios[a] and Maree Pather[b]

[a, b] Nelson Mandela Metropolitan University

[a, b] Faculty of Engineering, Department of Computer Studies, PO Box 77000, Nelson Mandela Metropolitan University, Port Elizabeth, 6013

[a] demetriosl@gmail.com

[b]maree.pather@nmmu.ac.za

ABSTRACT

The need for organisations to share data and collaborate on a large scale with geographically dispersed parties has increased dramatically in recent years. Grid Services allow for large scale collaboration between geographically-dispersed parties running diverse hardware and software platforms, over public networks such as the Internet. Grid Services are an evolution of Web Service technology and other open, platform-independent standards. Current research efforts have been undertaken to standardize grid implementations. With the efforts of the Global Grid Forum (GGF) and other interested parties, the Globus Toolkit has been developed. The focus of this paper is to define a holistic security strategy for implementing Globus-based Grids.

The Globus Toolkit is an open source software initiative, providing a set of tools and a platform for grid developers to build onto. The Toolkit is currently the de facto standard for Grid Service implementations, and is in its fourth major revision GT4 (Globus Toolkit version 4). The Globus Toolkit consists of a number of core components for implementing grids; the component of interest to this research is the Globus Security Infrastructure (GSI). This research looks at a layered approach to securing grids, making use of a defence-in-depth approach. The

focus is on the Globus Toolkit and GSI, local hardware and software configurations for remote sites, and communications (i.e. TCP/IP stack, RMI, RPC, etc). The STRIDE model will be used to provide a base for understanding hackers attack methodologies and threats faced by modern Grids.

KEY WORDS:
Grid Security, Globus, Brokered Grids, STRIDE, Grid interoperability

A BROKERED APPROACH TO INTEROPERABLE SECURITY IN OGSA-BASED GRID SYSTEMS

1 INTRODUCTION

Due to a number of factors, grid computing has gained in popularity and application. To date many distributed computing paradigms exist, such as Common Object Request Broker Architecture (CORBA), Java's' Remote Method Invocation (RMI), Common Object Model (COM), Web services, etc. Grid Services are an evolution on existing paradigms (Foster, C. Kesselman, S. Tuecke; 2001). The use of open standards such as Open Grid Service Infrastructure (OGSI), extensible Mark-up Language (XML) and Simple Object Access Protocol (SOAP) easily allows for heterogeneous platforms to communicate and share computing resources within a virtual organisation (VO) context.

According to Foster, the goal of Grid Computing is "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations". Sharing is not just denoted as file exchange or just data sharing, but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. (Foster, et al;2001). Furthermore, a set of individuals and/or institutions defined by such sharing rules form what is known as a virtual organization (VO). Current implementations of grid computing models have had great success in a variety of contexts, from the monitoring of natural phenomena, to the prediction of market trends among consumers, to name a few. However, as the face of modern computing evolves, so do the challenges to the underlying technologies that drive it.

These challenges include: location, connectivity and platform configurations. Implicit in these challenges are issues of interoperability, ownership and responsibility, security, performance, and reliability.

Grid Services is a distributed computing paradigm, built on Web-Services and SOAP. The use of XML Web-Services as an underpinning technology – notably the WS-* set of specifications for extending SOAP functionality - makes it possible for most of these issues to be addressed (Foster, et al; 2001). A Grid is a collection of Grid Services, or other Grids logically grouped into a Virtual Organization (VO). Grid Services provide a number of services, including processing or computational power, database housing, application hosting and sharing.

The primary focus of this research is to provide a generic and coherent security framework, to protect Grid Computing resources and users from hackers and intrusion attacks. The process of defining a detailed security strategy for all known vulnerabilities, attacks, possible variants on know attacks, and new or unknown attacks can be a daunting task; almost impossible, at the rapid rate of availability of new hacking tools. It might be more economical to typify the hackers' intentions and generic goals when attacking a system, with a view to defining a threat model that can be applied to Grids and Grid services.

A two-level strategy will be discussed in implementing a defence-in-depth strategy for protecting Grids, within the Globus Context. (The Globus Toolkit is used for developing Grid Service solutions; see http://www.globus.org/toolkit/). The first level is concerned with Grid Services (the lowest level in a Grid), while the second level will look at the Grid as a whole and the particular challenges faced by Grid designers when implementing them. The STRIDE model (Meier et al; 2003) for hacker behaviour will be investigated and applied to a risk assessment methodology, to provide Grid designers with a framework for developing security policies to protect their Grid Services. Additional information will be provided on Grids and the Globus Toolkit, as well as a threat-modelling strategy that can be applied to Grid computing.

To summarise, this paper will:

- Discuss Grid Computing and supporting technologies, such as the Globus Toolkit,

- Provide an introduction to the STRIDE model for threat-modelling, and
- Describe a holistic security framework for defining security strategies in a Grid environment.

2 GRID COMPUTING

Grid Computing allows large heterogeneous groups to share computing processing power, as well as other computing resources. Foster defines a grid as follows: "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." (Foster, 2002). Foster, furthermore, proposes a three-point checklist, to which grid systems must comply:

1. Coordinates resources that are not subject to centralized control;
2. Using standard, open, general-purpose protocols and interfaces; and
3. To deliver nontrivial qualities of service.

Grid research is currently focused on standards to facilitate resource virtualization and to accommodate intrinsic heterogeneity of resources in distributed environments (Stuer, V. Sunderam, J. Broeckhove ; 2004). The concept of Grid Services is a natural evolution on Grid Computing.

Open Grid Services Infrastructure (OSGI) is a specification which defines basic mechanisms and interfaces which can be used to build Grid functionality. Open Grid Services Architecture (OGSA), is an open standard for Grid Services implementation. Standard frameworks, based on XML, are being used to describe standard service specifications, to allow clients to discover and use services across platform, and domain contexts (Ibid). OGSA defines a best practice for implementing grid-enabled services.

The Globus Toolkit (http://www.globus.org/Toolkit/), now in version 4, is an open source software framework, designed to implement grid services. Its goal is to develop and promote standard grid protocols to enable interoperability and shared

infrastructure. A lot of the work done on the Globus project is through the Global Grid Forum (GGF).

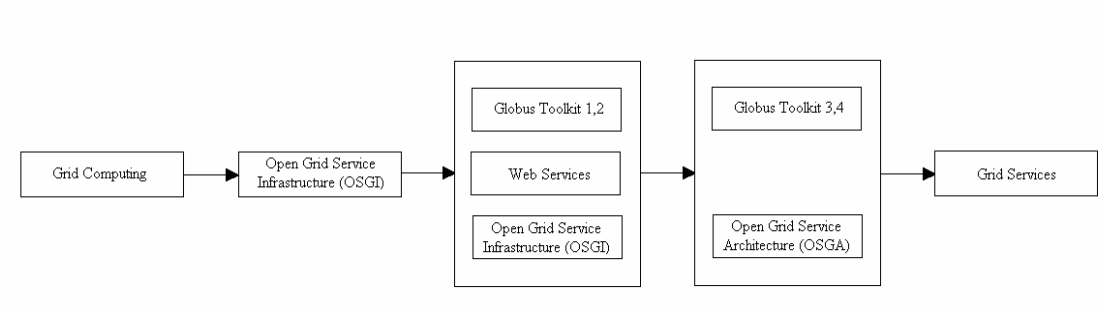

Figure 1: Timeline diagram of Grid Services, concepts and related standards and technologies

The above diagram shows a logical timeline of standards and technologies that support or make up modern Grid Services, the following section will discuss the Globus Toolkit in more detail.

2.1 Globus Toolkit

The Globus Toolkit facilitates an open source implementation of all the protocols and primitives defined by Open Grid Service Infrastructure (OSGI), for implementing grid services (Sandholm; 2003). The Toolkit consists of a number of components, allowing one to develop and implement a grid service. This section will introduce these components and briefly discuss them.

The Globus Toolkit has a layered architecture; high level global services are built on a core set of lower level services. At the bottom of the hierarchy, and possibly one of the most important services, is the resource management service, Globus Resource Allocation Manager, or GRAM; this is responsible for assigning as well as de-allocating resources to services. (Foster, 1998).

In most distributed system architectures, communication plays a key role. The Globus Toolkit provides a communication component, NEXUS. NEXUS is a library of lower level communication APIs that provide support for higher level communication (Foster, 1998).

Security is also a major concern in grid implementations. Security needs in grids are diverse, including authentication, access control and privacy. Globus Security Infrastructure (GSI) is the component within the Toolkit that implements security. GSI primarily looks at the problem of authentication, and therefore leaves open a large area for future research in the security space (Ibid).

In a dynamic environment such as in a grid system, the need to be able to easily access information about services, components, and applications, in a timely fashion, is important. This is in order to allow for adaptation to changes in system structure and state. Globus Meta-Computing Directory Service (MDS) stores and makes accessible information such as the architecture type; operating system version and amount of memory on a computer; network bandwidth and latency; available communication;

Protocols; and the mapping between IP addresses and network devices (Foster, 1998). MDS provides tools and APIs to allow for discovery, publishing and access information about the structure and state of a grid.

Health Beat Monitor (HBM) provides simple management services for monitoring the health and status of sets of remote processes. The HBM consists of several client APIs. A process can register with the HBM, which then acts as a data-collection base, periodically receiving “heart-beat” information about a process. Other processes can query the HBM for the status of another process.

Globus also provides Global Access to Secondary Storage (GASS), a component that allows programs with access to simple C I\O libraries the ability to open, edit and save files on remote computers.

The final core service in the Globus Toolkit is Globus Executable Management (GEM). GEM supports the remote identification, creation and location of executables in heterogeneous environments.

Grid Concepts and the Globus Toolkit were discussed in this section, the following section will discuss threat modelling and hacker behaviour.

3. Threats

To understand the importance of securing one's information, it is important to understand what are the threats and impact associated with insufficient security (Whitman, 2003). A wide range of threats exist. These threats are unique for the various parts of a grid, although the attacker's (generic) goals might be the same (Meier, Mackman, Vasireddy, Dunner, Escamilla, Murukan; 2003). Knowing how and why a hacker can attack an information system is a good starting point to identifying threats to an organization's information assets.

3.1 Attackers goals

There are a wide range of possible attacks, and further fine-grained variations on these attacks. The best method to classify threats to one's system is to identify the hacker's goals when performing an attack. STRIDE is the acronym for an approach to categorize different threat types (Ibid):

- **Spoofing** - The hacker's goal when spoofing is to try gain access to the system by mimicking legitimate user-credentials or network traffic.
- **Tampering** – This is the unauthorized altering of information, while it is in transit between two computers.
- **Repudiation** – Prevents administrators from knowing if users (legitimate or not), have performed an action.
- **Information disclosure** – This is the unwanted exposure of private information.
- **Denial of Service** – This is the process of making services un-available to users.
- **Escalation of privileges** – This attack occurs when a user of limited privileges assumes the roll of a privileged user, in order to steal, corrupt, or deny access to information asset.

3.2 Hacker's methodology

Microsoft (Ibid) identifies the basic attack approach adopted by hackers, this approach defines generic steps a hacker will need to perform in order to complete a successful attack; not all steps are required in every instance. The steps in the hacker's methodology are listed below:

1. **Survey and assess** - Survey and assess is the initial stage of the hacking process. The hacker will try to learn of possible servers and services on the network. The hacker will then try to find possible weakness and exploits, to try and gain access to the target machine.
2. **Exploit and penetrate** - Once the hacker completes the survey phase, the next step is to exploit and penetrate the target.
3. **Escalate privileges** - Upon completing the attack and delivering the payload, the hacker will then attempt to create a backdoor to access the desired server. Immediately an attempt will be made to escalade privileges, specifically to administrator.
4. **Maintain access** - Once the attacker has administrative privileges, they will try and make further access easier and try to hide his or her tracks. A common method of making back door access possible is to plant back-door applications. Hackers will often attempt to clear event logs at this stage.
5. **Deny service** - If the attacker is not successful in his or her attack, they will try launch a Denial of Service attack (DoS), to deny others use of the service.

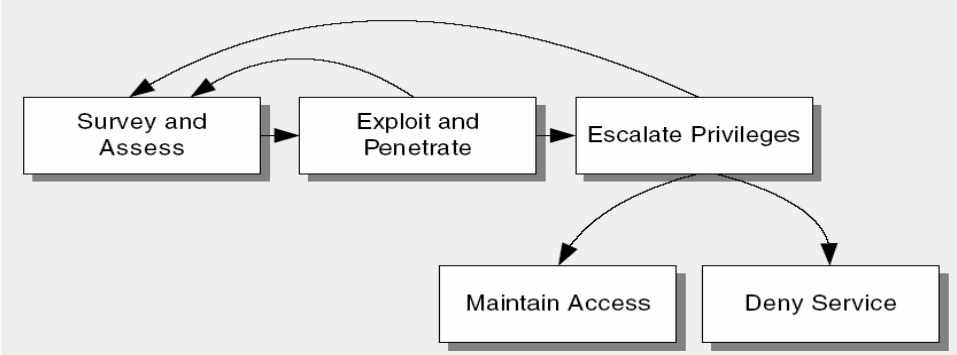

Figure 2. Steps in a typical attack (Meier et al, 2003)

In a Grid Community, nodes (clients, servers, brokers), messages and message pathways are exposed to a range of threats.

4. Holistic Grid Security Framework

Grids can be logically divided into two levels, based on security needs and challenges. The first layer is the local grid service layer, which is concerned with a data or computational grid service as a separate local entity below the common grid infrastructure in the VO (Virtual Organization) context. The second layer is the Common Grid layer. This layer consists of all GT grid services. A single VO can span countries, or the globe.

One of the biggest problems faced by Grid designers is implementing authentication and authorization between Grid Services or sites. Each site may have its own local security policy, and will make use of a different set of technologies (Foster; 1998b). This includes security issues when crossing trust domains and grid-to-grid security issues, such as single-sign on authentication and authorization. We will first look at the lowest level, the local grid services that make up a grid. Consider the following diagram:

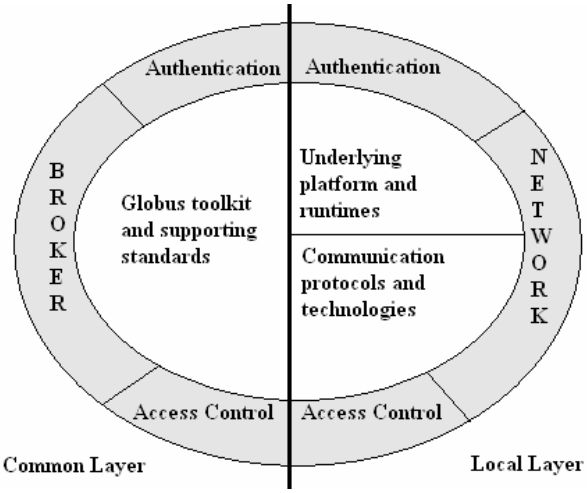

Figure 3: Logical view of a Grid

In the above diagram, the core represents a local Grid Service. The outer layer represents aspects and challenges of a Grid in a VO (virtual organization) context. Local Grid services can be divided into four logical security layers:

Local (Grid Service) level:

- **Underlying platforms and runtimes:** Platform security is based on the hardware and software platforms the Grid Service is hosted on, i.e. INTEL x86, SPARC, etc. running LINUX, UNIX or Windows operating systems. Runtimes dictate the security runtime environment; the grid software is typically hosted in e.g. JAVA or .NET.
- **Communications protocols and technologies:** Service components can communicate over a variety of mediums and protocols. The most commonly used communication protocol is TCP/IP, over variety of mediums, broadband, wireless, VPNs, etc. Each communication mechanism involves an appropriate local security implementation, e.g. IPSec.
- **Network:** the network "cloud" between (local and) grid participants, and (local and) grid services in a virtual organization is, fortunately, commonly based on TCP/IP, with interoperability on LAN and WAN interfaces being provided for through hardware and software gateways.
- Implicit in the **actual service components** will be additional security measures such as .NET strong names (with embedded credentials) and role-based security at the component/class/method level.

The Common Grid Layer Challenges:

- **Authentication:** A variety of authentication challenges are presented when multiple sites or grid participants have varying security policies and authentication implementations. Identification and authentication credentials have to be persisted from the common layer to local layer.

- **Access Control (Authorization):** The challenge of maintaining access control assertions down to the local components is obviously great as common policies and interoperable implementations are critical factors. Hence, Web Services standards, such as WS-Security and WS-Policy are crucial to the GT framework.

- **Broker:** An abstracted software component, acting as an intermediary between parties, is the backbone of Grid Services integration. It deploys GT (or equivalent) middleware, common security and interoperability policies and implementations. This layer provides access to the local grid service and associated services (database, application hosting, processing, etc). It uses interoperable standards, such as XML and SOAP. It uses GT mechanisms for mapping security credentials and interoperability mechanisms from the Common Grid Layer to the Local Grid Layer.

Next, Grid Services security will be considered in terms of the STRIDE approach discussed earlier.

4.1 Grid Services Security and STRIDE

The STRIDE model for threat modelling was introduced previously, as well as a hacker methodology for attacking information systems.

- Grid services have unique security needs, largely due to their open nature and interconnectivity. (Grid services are largely un-standardised, in terms of underlying platforms and communications technology. As discussed, they are often built on a variety of hardware, software, and operating system platforms, as well as a range of possible communication protocols and technologies. (Baker, et al; 2000). However, the common layer is, typically, standardized in terms of using open standards, such as XML and SOAP, and the Globus Toolkit (a de facto standard for building Grid services). This layer is standardized in order to facilitate integration of underlying heterogeneous platforms and technologies.

In the previous section it was determined that there are several generic hacker goals, as well as a set of generic steps a hacker will follow to attack a system. The following table shows what goals are typically applicable to each particular layer, defined above, in a Grid service VO.

	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service (DoS)	Escalation of privileges
Underlying platform and runtimes	X		X		X	X
Communication protocols and technologies	X	X			X	
Local Grid Service	X			X	X	X
Broker: Globus Toolkit and supporting standards	X		X	X	X	X

Figure 4: table of typical goals of a hacker when attacking each layer of a grid

The above table can be used as a generic basis for developing a local security strategy to protect a Grid Services Deployment. The details of implementation are beyond the scope of this paper. However consider the following scenario:

Tampering is a risk associated with the communications layer of a grid. A typical method of tampering with network traffic is a “man-in-the middle” attack, in which a hacker will intercept traffic in transit from one node, read the contents and alter it, then pass it on to the intended recipient. A number of controls can be implemented to combat this threat, such as encryption. On a high level, we have determined that the grid implementation will require encryption to protect information in transit. When deciding on a Grid-wide encryption strategy for information in transit, we can determine if IPSec will be used, or more commonly in this instance, encrypted SOAP packets at layer 7 (of the OSI reference model).

4.2 Grid Security Implementation Scenario

Grids require standard security functions, such as, authentication, access control, integrity, privacy, and non-repudiation (Foster, 1998b). This is difficult to implement in a Grid-wide Community, due to a number of factors. VOs (Virtual Organizations) can be made up of a number of diverse geographically disperse sites, implementing non-compatible local security policies\technologies. Consider the following scenario:

User-A at Site-A starts an analysis program that sends code to be executed on Site-B, but Site-B requires a dataset on Site-C to perform the analysis. The application at Site-A contacts a broker at Site-D to obtain idle resources needed to process the task at hand. The Broker then initiates communications with sites E,F,G in order to complete the task at hand. These sites will need to maintain communication between them (possibly using a multicast protocol), as well as the broker, the original site (requesting site), and the user.

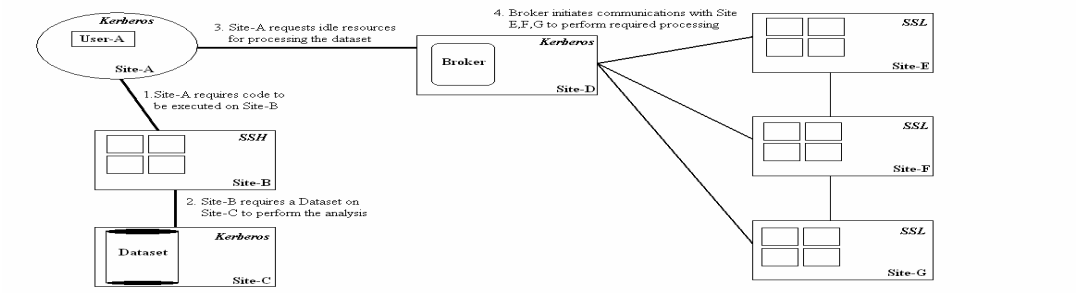

Figure 5: Example of large scale distributed computing environment

The above scenario depicts many distinctive characteristics of the Grid Computing environment (Foster, 199b):

- The user population is large and dynamic.
- The resource pool is large and dynamic.
- A computation (or processes created by a computation) may acquire, start processes on, and release resources dynamically during its execution.

- The processes constituting a computation may communicate by using a variety of mechanisms. Low level communications (e.g. TCP/IP sockets) can be created and destroyed dynamically during program execution.
- Resources may require different authentication and authorization mechanisms and policies, which we will have limited ability to change. In the above example, this was illustrated this by showing the local access control policies that apply at different sites. These include Kerberos, Secure Socket Library (SSL) and Secure Shell (SSH).
- An individual user will be associated with different local name spaces, credentials, or accounts, at different sites, for the purposes of accounting and access control.
- Resources and users may be located in different countries.

There is need to provide security solutions to grid users that can allow computations, such as in the above described scenario. These solutions must allow for the co-ordination of diverse access control policies and to allow them to operate securely in heterogeneous environments (Foster, 199b).

In order to achieve a global security infrastructure within a VO, a broker can be used to facilitate communications, authentication and authorization at a central site. The implementation of various services and middleware can allow for this.

Grid users are provided two sets of credentials, one applicable to their local security policy and another to a global Grid security policy. A broker service can be used to maintain a table of mappings for user credentials, which allows for comparison to a global security policy for access to resources (Foster, 199b). This mapping of user credentials can provide a transparent single sign-on to the user when interacting with the grid.

1. A user provides the credentials needed to log onto the grid.
2. The user initiates a process that requires remote grid resources.
3. The user's grid credentials are tested against a global access-control policy.
4. The user's rights to that resource are determined.

5. If they have sufficient rights, the grid initiates the communication and provides that grid service with the correct level authentication.

This sort of policy can be implemented using a group policy structure. The structure consists of several elements:

- Resource
- Users
- Groups
- Privilege

A resource is defined, groups are linked to a resource, and one group can contain many users. A group then has a privilege to access the resource defined, one resource can have many groups linked to it.

This simple paradigm can allow for complex authorization structures though out the grid and can cater for scalability. However there are some complexities involved in the implementation and maintenance of the proposed structure. Middleware can be used to reduce the complexities of maintaining the proposed structure, however this falls outside the scope of this research.

A holistic grid security structure was investigated, and it was found grids can be divided into two logical layers, the grid layer and grid service layer. Each of these defined layers has their own security needs. A framework to implement a security strategy was described.

5. Conclusion

A layered approach to securing grids was introduced in this paper. Grids provide a powerful mechanism for collaboration and sharing data and processing resources. The Globus Toolkit was briefly discussed, the Toolkit provides Grid designers a standardised set of software tools and libraries for implementing grid services, and is considered the de facto standard for implementing grid services. STRIDE was discussed as a threat model for categorizing hacker's action and behaviours, based on the outcome of the attack performed or the hacker's goal in attacking a Grid.

The anatomy of a Grid and Grid Services were discussed. It was suggested that Grids can be divided into two logical layers, the common and local layers, each with its own security needs. The lower of the two layers the local layer is concerned with security at a single site. The higher level, or the common layer, is concerned with "global" Grid security issues, including authentication and authorization between Grid Services sites.

A security strategy taking into account all possible attacks and hacks against a Grid is a daunting task. It was proposed here that STRIDE be used by Grid designers as a basis to develop security strategies to protect Grid Services. Each category of STRIDE was found to be applicable to aspects of a Grid Service, providing Grid designers a suitable framework for developing tailored Grid security strategies.

A brokered approach for providing authentication and authorization services on a common grid layer was discussed. Although this approach provides a means to solve the problem of single-sign authentication, grid-wide authorization, etc. It does require the use of Globus (or other standard) middleware. The complexities of implementing and maintaining a brokered approach provide an area for further research.

6. REFERENCES

I. Foster, C. Kesselman, "*The Globus Project: A status report*", (1998a)

I. Foster, C. Kesselman, G. Tsudik, S. Tuecke*, "A Security Architecture for Computational Grids",*(1998b)

M.Baker, R. Buyya, D. Laforenza, "*The Grid: International Efforts in Global Computing*", (2000)

I.Foster, C. Kesselman, S. Tuecke, "*The anatomy of the Grid*", (2001)

I. Foster, "*What is the Grid? Three point checklist*", (2002)

M.E. Whitman, "*Enemy at the Gate: Threats to Information Security*", (2003). Communications of the ACM.

J.D. Meier, A. Mackman, S.Vasireddy, M.Dunner, R. Escamilla and A. Murukan, "*Improving web security, threats and countermeasures*", (2003). Microsoft Pattern and Practices.

T. Sandholm, J. Gawer, "Globus Toolkit 3 core – A Grid Service Container Framework", (2003)

G. Steur, V. Sunderam, J. Broeckhove, "*Towards OGSA Compatibility in Alternative Metacomputing Frameworks"*, (2004)

References

Aiken, B., Strassner, J., Carpenter, B., Foster, I., Lynch, C., Mambretti, J., Moore, R., & Teitelbaum, B. (2000) "*Network Policy and Services: a Report of a Workshop on Middleware"*. RFC. RFC Editor.

Bellovin, S. M. (1989) "Security problems in the TCP/IP protocol suite". *ACM SIGCOMM Computer Communication Review archive,* 19(2), 32-48.

Berman, F., Fox, G.C., Hey A.J.G. (2003) "*Grid Computing - Making the Global Infrastructure a Reality"*. New York: John Wiley & Sons, Inc

Brebner, P., & Emmerich, W. (2005) "Two ways to grid: the contribution of Open Grid Service Architecture (OGSA) mechanisms to Service-centric and Resource-centric lifecycles". *Journal of Grid computing*, *4*(1), 115-131.

Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J. & Welch, V. (2002) "Design and Deployment of a National-Scale Authentication Infrastructure". *IEEE Computer, 33*(12), 60-66.

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001) "Web Service Description Language (WSDL) 1.1". *World Wide Web Consortium (W3C),* Retrieved November 15, 2006, from http://www.w3.org/TR/wsdl

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002) "Unraveling the Web services Web: An introduction to SOAP, WSDL, and UDDI". *IEEE Internet Computing*, *6*(2), 86-93.

Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Maguire, T., Snelling, D., & Tuecke, S. (2004) "*From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution*". Open Grid Forum, Retrived November 25, 2006, from http://www-128.ibm.com/developerworks/library/ws-resource/ogsi_to_wsrf_1.0.pdf

Czajkowski, K., Ferguson, D.F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., & Vambenepe W. (2004b) "The WS-Resource Framework version 1.1", *Open Grid Forum*. Retrieved July 31, 2006, from http://www.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf

Dierks, T., & Rescorla, E. (2004.) "*The TLS Protocol Version 1.1*". RFC 2246.

Ethereal, (2006), "Ethereal homepage". Retrieved 30 November, 2006, from http://www.ethereal.com/

Foster, I., & Kesselman, C. (1998). "*The Grid: Blueprint for a New Computing Infrastructure*", San Francisco: Morgan Kaufmann Publishers Inc.

Foster, I., & Kesselman, C. (1998b) "*The Globus Project: A status report*". Heterogeneous Computing Workshop, *7*(March), 4-18.

Foster, I., Kesselman, C., Tsudik, G., & Tuecke, S. (1998) *"*A Security Architecture for Computational Grids*", In Proceedings of the 5th ACM Conference on Computer and Communications Security* (San Francisco, California, United States, November 02 - 05, 1998). CCS '98. ACM Press, New York, NY, 83-92.

Foster, I. (2000) "Internet Computing and the Emerging Grid". Retrieved November 25, 2006, from: http://www.nature.com/nature/webmatters/grid/grid.html

Foster, I., Kesselman, C., & Tuecke S. (2001). "Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal for Supercomputing*, 15(3), 200-222.

Foster, I. (2002) "*What is the Grid? A Three Point Checklist*", Retrieved February 9, 2006, from http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf

Foster, I., Kesselman, C., Nick, J.M., & Tuecke S. (2002). "The Physiology of the Grid: the Open Grid Service Architecture for Distributed System Integration", *Open Grid Forum*. Retrieved July 31, 2006, from http://www.gridforum.org/ogsiwg/drafts/ ogsa_draft2.9_2002-06-22.pdf

Foster, I., Kesselman, C., Nick, J.M., &Tuecke, S. (2002b) "Grid services for distributed systems integration", *IEEE Computer*, 35(6), 37-46.

Foster, I., & Iamnitchi, A. (2003) "On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing"; *In Proceedings of the 2nd International Workshop on Peer-to-Peer Systems* (IPTPS '03), 118-128.

Foster, I., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., & Von Reich, J. (2006). "OGSA Specification 1.5", *Open Grid Forum*. Retrieved July 31, 2006, from http://fisheye.globus.org/viewrep/~raw,r%3D1.1/GlobusToolkit/gridservices/ docs/whitepaper.doc

Frier, A., Karlton, P., & Kocher, P. (1996) "The SSL 3.0 Protocol". *Netscape Communications Corporation*, Retrieved October 25, 2006, from http://wp.netscape.com/eng/ssl3/ssl-toc.html

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995) "*Design Patterns: Elements of Reusable Object-Oriented Software"*. Addison-Wesley Longman Publishing Co., Inc.

Gerndt, M. (2004) "Grid Computing: An Infrastructure for Large-scale Simulations". Retrieved on November 25, 2006, from http://www.lrr.in.tum.de/~gerndt/home/Vita/Publications/04GridComputingJordanien.pdf

Globe, C., De Roure, D. (2002) "The Grid: An application of the semantic web". *ACM SIGMOD Record*, *31*(4), 65 – 70.

Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.J., & Nielsen, H. F. (2003) "SOAP Version 1.2 Part 1: Messaging Framework", *World Wide Web Consortium (W3C)*. Retrieved October 15, 2006, from http://www.w3.org/TR/soap12-part1/

Heckman, R. (2006). "Attack Modeling vs Threat Modeling". *Builder AU*. Retrieved 21 November 2006 from: http://builderau.com.au/blogs/intothebreach/soa/Attack_Modeling_vs_Threat_Modeling/0,339027621,339243880,00.htm

Hernan, S., Lambert, S., Ostwald, T., & Shostack, A. (2006) "Uncover Security Design Flaws Using The STRIDE Approach". *MSDN Magazine*, *21*(12). Retrieved 2 November, 2006 from: http://msdn.microsoft.com/msdnmag/issues/06/11/ThreatModeling/default.aspx

Houle, K.J., & Weaver., G.M. (2001) "Trends in denial of service attack technology". CERT coordination center, Retrieved November 15, 2006, from http://www.cert.org/archive/pdf/DoS_trends.pdf

Howerd, M., & Lipner, S. (2003) "Inside the windows security push". *IEEE Security and Privacy magazine*, 1(1), 57-61.

Humphrey, M., & Thompson, M. R. (2005) "Security for grids". *Proceedings of the IEEE*, *93*(3), 644-652.

Kendall, K. E. & Kendall, J. E. (2002) "*System analysis and design*", Prentice-Hall, Inc

Krawczyk, H., Bellare, M., & Canetti, R. (1997) "*HMAC: Keyed-Hashing for Message Authentication*". RFC 2104

Kontzer, T., & Whiting, R. (2004) "Industry Sees Growth in Grid Computing". *Information week*, Retrieved 5 December, 2006, from http://www.informationweek.com/story/showArticle.jhtml?articleID=23901953

Kusnetzky, D., & Olofson, C. W. (2004) "*Oracle 10g: Putting Grids to work*". Oracle Corporation.

Li, M. C., Cui, Y., Tian, Y. (2006) "A New Architecture of Grid Security System Construction". *Proceedings of the 2006 International Conference on Parallel Processing Workshops*, (ICPPW'06), 100-108.

Liming, L., Garritano, T., & Tuecke, S. (2004) "Standardizing the Grid". *Cluster World magazine*, *2*(5), 2-4.

Meier, J.D., Mackman, A., Vasireddy, S., Dunner, M., Escamilla, R., & Murukan. A., (2003). *"Improving web application security: threats and countermeasures*", Microsoft Patterns and Practices.

Nagaratnam, N., Janson, P., Dayka, J., Nadalin, A., Siebenlist, F., Welch, V., Foster, I., Tuecke, S. (2002) "The security architecture for open grid services". *Open Grid Forum Security Group*, Retrieved October 20, 2006, from http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/OGSA-SecArch-v1-07192002.pdf

Microsoft. (2006). "Microsoft security glossary". Microsoft Corporation, Retrieved 23 November 2006 from: http://www.microsoft.com/security/glossary.mspx#b

NIST (2001) "Federal Information Processing Standards Publication 197". *National Institute for Standards and Technology (NIST)*, Retrieved 30 November, 2006, from http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

MIT Technology Review, (2003) "Ten emerging technologies that will change the world", *The MIT Enterprise Technology Review,* February 2003.

Open Grid Forum (OGF). (2005) "A Globus Primer or, Everything You Wanted to Know about Globus, but Were Afraid to Ask: *Describing* Globus Toolkit Version 4*". Open Grid Forum*, Retrieved September 4, 2006, from http://www-unix.globus.org/ toolkit/docs/4.0/key/GT4_Primer_0.6.pdf

Ramakrishnan, L. (2004) **"**Securing next-generation grids". *IT Professional*, 6(2),34- 39.

Rivest, R., Shamir, A., Adleman, L. M. (1978) "A method for obtaining digital signatures and public-key cryptosystems". *Communications of the ACM,* 21(2), 120–126.

Salter, C., Saydjari, O., Schneier, B., & Wallner, J. (1998) "Toward a Secure System Engineering Methodology". Retrieved November 15, 2006, from http://www.schneier.com/paper-secure-methodology.pdf

Sandholm, T., & Gawer, J. (2003) "Globus Toolkit 3 core: A Grid Service container framework", *Open Grid Forum*. Retrieved February 16, 2006, from http://www-unix.globus.org/toolkit/3.0/ogsa/docs/gt3_core.pdf

Schneier, B. (2000) "*Secrets & Lies: Digital Security in a Networked World*", John Wiley & Sons, Inc.

Schopf, J.M., & Nitzburg, B. (2002) "Grids: The top ten questions". *Scientific programming*, *10*(November), 103-111.

Schulze, B., & Madeira, E.R.M. (1997) "Contracting and Moving Agents in Distributed Applications: Based on a Service Oriented Architecture". *First International Workshop on Mobile Agents* (April 1997), 74-85.

Siebenlist, F., Welch, V., Tuecke, S., Foster, I., Nagaratnam, N., Janson, P., Dayka, J., & Nadalin, A. (2002) "Global Grid Forum Specification Roadmap towards a Secure OGSA", *Open Grid Forum.* Retrieved December 1, 2006, from www.globus.org/toolkit/security/ogsa/draft-ggf-ogsa-sec-roadmap-01.doc

Surridge, M. & Upstill, C. (2003) "Grid Security: lessons for peer-to-peer systems". *IEEE Proceedings of the Third International Conference on Peer-to-Peer Computing*, *3*(September), 2-6.

Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S., & Siebenlist, F. (2004), "X.509 Proxy Certificates for Dynamic Delegation", *Proceedings of 3rd Annual PKI R&D Workshop*, *3*(April), 42-58.

Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L., Tuecke, S. (2003) "Security for grid services". *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, *12*(June), 48-57.

Whitman, M.E. (2003). "Enemy at the Gate: Threats to Information Security". *Communications of the ACM*, *46*(8), 91-95.

Whitman, M.E., & Mattord, H.J. (2003). "*Principles of Information Security*". Thomson Course Technology.

Wikipedia, (2006), “Software cracking”. *Wikipedia Foundation, Inc*, Retrieved on 5 December 2006 from: http://en.wikipedia.org/wiki/Software_cracking