Information-Security Use-Cases for Grid Architectures

Status of This Memo

This memo provides information to the Grid community on common information security use-cases. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2007). All Rights Reserved.

Trademarks

OGSA is a trademark of the Open Grid Forum.

DRAFT

1 Introduction

This document serves to identify common use-cases for aspects of information security within an open grid architecture.

The identification of operational use-cases plays an important role in deriving functional requirements, which in turn suggest design and implementation mechanisms. Use-cases are traditionally human-centric: they define actions between external actors (users) and the system (which is typically treated as a black box) to achieve specific business goals or tasks. Use-case descriptions generally strive to avoid implementation-specific language, focusing more on the purpose and properties of how the actor(s) and system(s) interact.

The reality, however, is that the relationship between use-case-identification and mechanism-design is not unilateral; we often find that the prevalence of existing or legacy technologies in fact drive use-cases and system requirements. In this document, we will present both fundamental use-cases (processes and their abstract properties) and use-cases that are mechanism-specific (processes whose environments leverage specific technologies whose mechanisms must be considered).

2 Fundamental Use-cases

The information security of a protocol or system is a set of related but somewhat independent properties that can be categorized into two overlapping concerns: communication security and system administration security. The overlap is a necessary one: communications are carried out by systems and access to systems is through communications channels.

The fundamental secure communication use-case is an interaction scenario between two networked parties (e.g., a client and a service) over an insecure communications channel. To be more concrete, let's examine the simple scenario in which a grid client *A* wishes to read the contents of a remote data source *B* (e.g., a federated grid file) over an insecure network (i.e., a network having the properties of the Internet Threat Model as defined in RFC 3552). We give simple use-cases for the following secure communication properties:

i. *Authentication*: *A* and *B* wish to ensure that they are indeed communicating with each other (instead of imposters).

ii. *Confidentiality*: *A* and *B* do not wish to expose any information regarding the read request or the returned data to third-parties.

iii. *Integrity*: *A* and *B* do not wish that the either the read request or the returned data be subject to modification while in transit

We continue this simple "data-read" scenario to illustrate use-cases for the following system security properties:

i. *Authorization*: The service wishes to only provide data-read access to a limited set of users (i.e., those users who can demonstrate specific identities or attributes). In order to curb inappropriate usage even by acceptable actors, the decision to allow access may incorporate arbitrary service-specific policy. (E.g., "you may submit jobs to the job queue at a maximum frequency of once every 60 seconds".)

ii. *Non-repudiation & Auditing*: *A* and *B* wish to be able to demonstrate to a third party that the information they received from the other cannot be denied later.

For more information on the above security properties or the Internet Threat Model, consult the IETF RFC 3552.

DRAFT

3 Mechanism-specific Use-cases:

This section presents use-cases for simple communication patterns. As in the previous section, we specify properties of the environment in which our use-cases are manifested. These properties in this section, however, are mechanism-specific: we describe our assumptions about the technologies and mechanisms that will manifest themselves in these communication patterns and how they factor into the use-case's interactions.

The communication environment that we assume is one in which grid components communicate via SOAP messages. These messages are conveyed over a network transport protocol such as HTTP/HTTPS/JMS/etc. Except where noted, our communication pattern use-cases assume HTTPS at the network transport level. The goal of any implementation suggested by the use-cases below would be to preserve the abstract information security properties from the previous section. As such, we include considerations for these use cases that such an implementation would need to address.

i. *Simple one-way communication pattern.* A message is delivered from one grid component to another without necessity of a response. This pattern is frequently manifested in scenarios that employ notification mechanisms; consider an example of lifetime-notification in a metadata repository wishes to be notified upon the termination of a basic execution service that it monitors.

Depending on the one-way-ness of the transport mechanism, the client may not be able to use "handshake" information in order to authenticate the service and be assured of confidentiality. For example, consider the use-case in which a subscriber *must* receive notification of an event (such as a service migration event), even in the presence of intermittent network failures. An implementation that this use-case suggests is one in which a reliable messaging transport such as JMS would be employed to ensure message delivery. A handshake (such as the SSL/TLS handshake) for providing the client with key and trust-based cryptographic identity may not exist, causing this use-case to possibly depend on an external mechanism for key distribution and trust.

ii. *Simple request-response communication pattern.* A message is delivered from one grid component to another with the necessity of a response. A bidirectional transport protocol such as HTTPS is a good fit for this pattern and is an expected mechanism-specific property of this use-case. As such, any X.509 certificates that are communicated during SSL/TLS handshake that do not include the service's network address (possibly because it may vary) are not sufficient to authenticate the service identity to the communication endpoint. Therefore this use-case suggests an external mechanism for key distribution and trust.

iii. *Delegation*. The response data is dependent on communication that must be performed with other grid services. This is a superposition of the above two communication patterns. Consider the job-submission scenario in which jobs are submitted to a queuing service which must further delegate job instantiation to basic execution services (which in turn may need to further delegation actions to with file, data, and application deployment services).

A restricted version of this use-case is one in which the caller employs a one-way communication to initiate the process. The one-way pattern allows the caller to operate in a network environment that does not allow incoming messages from third-parties or allows it to terminate before the entire process has completed. In these cases, delegation protocols that require callbacks (e.g., certificate signing requests) may not be feasible.

iv. *Communication with intermediaries*. Consider again the scenario in which one wants reliable delivery of lifetime notification messages. This scenario requires an end-to-end security solution. One implementation suggested by this use-case is the use of

DRAFT

reliable delivery functionality is provided at the transport-level. This would require message-level security. An alternative solution might provide reliable delivery functionality at the message-level (via first-class grid services). In this case, delegation (and the considerations that accompany it) and document-level encryption (i.e., message-level) would be required for information security.

v. *Communication using multiple identities/attributes.* Consider the scenario in which clients may need to communicate multiple security credentials to a service for authentication/authorization. More specifically, a delegation scenario may require that an intermediary need to communicate its own identity credential as well as a delegation credential. Another scenario is in which users obtain a set of credentials for the different administrative domains with which they will interact. (This relieves services from the burden of identity mapping.) Communicating with multiple identities may require message-level security. (Transport-level protocols such as SSL/TLS are specific to single X.509 credentials.)

vi. *Hosted service resources.* Due to prevalent Web services technology, it is likely that multiple stateful service resources will be hosted within a single web application container. Therefore transport-level communication (specifically authentication) occurs between the client and the container. A client may want to ensure that the container being communicated with actually contains a specific resource. This would require giving resources cryptographic identity, an external mechanism for distributing such key material, and trust-based message-level security for authenticating it.

Resource replication and migration are also use-cases that suggest an implementation in which resources are given cryptographic identity to ensure information security. Stateful resources that are replicated for high-availability (e.g., grid files) may be deployed within multiple containers, yet all copies should have the same trust-based cryptographic identity. Stateful service resources that are migratable (e.g., in response to container decommissioning or insufficient computing resources) should also maintain the same authenticatable identity regardless of hosting container.

Although it is not the stated purpose of a use-case document to derive mechanism or implementation, it is interesting to note that the considerations discussed for the above scenarios suggest that a sufficient implementation provide both an external mechanism for key distribution and trust as well as a mechanism for message-level security.

DRAFT