UNIT V

TRUST MODELS FOR GRID SECURITY ENFORCEMENT

Many potential security issues may occur in a grid environment if qualified security mechanisms are not in place. These issues include network sniffers, out-of-control access, faulty operation, malicious operation, integration of local security mechanisms, delegation, dynamic resources and services, attack provenance, and so on. Computational grids are motivated by the desire to share processing resources among many organizations to solve large-scale problems. Indeed, grid sites may exhibit unacceptable security conditions and system vulnerabilities.

On the one hand, a user job demands the resource site to provide security assurance by issuing a security demand (SD). On the other hand, the site needs to reveal its trustworthiness, called its trust index (TI). These two parameters must satisfy a security-assurance condition: TI ≥ SD during the job mapping process. When determining its security demand, users usually care about some typical attributes. These attributes and their values are dynamically changing and depend heavily on the trust model, security policy, accumulated reputation, self-defense capability, attack history, and site vulnerability. Three challenges are outlined below to establish the trust among grid sites .

The first challenge is integration with existing systems and technologies. The resources sites in a grid are usually heterogeneous and autonomous. It is unrealistic to expect that a single type of security can be compatible with and adopted by every hosting environment. At the same time, existing security infrastructure on the sites cannot be replaced overnight. Thus, to be successful, grid security architecture needs to step up to the challenge of integrating with existing security architecture and models across platforms and hosting environments.

The second challenge is interoperability with different —hosting environments.‖Services are often invoked across multiple domains, and need to be able to interact with one another. The interoperation is demanded at the protocol, policy, and identity levels.

For all these levels, interoperation must be protected securely. The third challenge is to construct trust relationships among interacting hosting environments. Grid service requests can be handled by combining resources on multiple security domains. Trust relationships are required by these domains during the end-to-end traversals. A service needs to be open to friendly and interested entities so that they can submit requests and access securely.

Resource sharing among entities is one of the major goals of grid computing. A trust relationship must be established before the entities in the grid interoperate with one another. The entities have to choose other entities that can meet the requirements of trust to coordinate with. The entities that submit requests should believe the resource providers will try to process their requests and return the results with a specified QoS. To create the

proper trust relationship between grid entities, two kinds of trust models are often used. One is the PKI-based model, which mainly exploits the PKI to authenticate and authorize entities; we will discuss this in the next section. The other is the reputation-based model.The grid aims to construct a large-scale network computing system by integrating distributed, heterogeneous, and autonomous resources. The security challenges faced by the grid are much greater than other computing systems. Before any effective sharing and cooperation occurs, a trust relationship has to be established among participants.

Otherwise, not only will participants be reluctant to share their resources and services, but also the grid may cause a lot of damage .

A Generalized Trust Model

At the bottom, we identify three major factors which influence the trustworthiness of a resource site. An inference module is required to aggregate these factors. Followings are some existing inference or aggregation methods. An intra-site fuzzy inference procedure is called to assess defense capability and direct reputation. Defense capability is decided by the firewall, intrusion detection system (IDS), intrusion response capability, and anti-virus capacity of the individual resource site. Direct reputation is decided based on the job success rate, site utilization, job turnaround time, and job slowdown ratio measured. Recommended trust is also known as secondary trust and is obtained indirectly over the grid network.

A general trust model for grid computing. Courtesy of Song, Hwang, and Kwok, 2005

Reputation-Based Trust Model

In a reputation-based model, jobs are sent to a resource site only when the site is trustworthy to meet users' demands. The site trustworthiness is usually calculated from the following information: the defense capability, direct reputation, and recommendation trust. The defense capability refers to the site's ability to protect itself from danger. It is assessed accoding to such factors as intrusion detection, firewall, response capabilities, anti-virus capacity, and so on. Direct reputation is based on experiences of prior jobs previously submitted to the site. The reputation is measured by many factors such as prior job execution success rate, cumulative site utilization, job turnaround time, job slowdown ratio, and so on. A positive experience associated with a site will improve its reputation. On the contrary, a negative experience with a site will decrease its reputation.

A Fuzzy-Trust Model

In this model , the job security demand (SD) is supplied by the user programs. The trust index (TI) of a resource site is aggregated through the fuzzy-logic inference process over all related parameters. Specifically, one can use a two-level fuzzy logic to estimate the aggregation of numerous trust parameters and security attributes into scalar quantities

that are easy to use in the job scheduling and resource mapping process.

The TI is normalized as a single real number with 0 representing the condition with the highest risk at a site and 1 representing the condition which is totally risk-free or fully trusted. The fuzzy inference is accomplished through four steps: fuzzification, inference, aggregation, and defuzzification. The second salient feature of the trust model is that if a site's trust index cannot match the job security demand (i.e., SD > TI), the trust model could deduce detailed security features to guide the site security upgrade as a result of tuning the fuzzy system.

Authentication and Authorization Methods

The major authentication methods in the grid include passwords, PKI, and Kerberos. The password is the simplest method to identify users, but the most vulnerable one to use.The PKI is the most popular method supported by GSI. To implement PKI, we use a trusted third party, called the certificate authority (CA). Each user applies a unique pair of public and private keys. The public keys are issued by the CA by issuing a certificate, after recognizing a legitimate user. The private key is exclusive for each user to use, and is unknown to any other users. A digital certificate in IEEE X.509 format consists of the user name, user public key, CA name, and a secrete signature of the user. The following example illustrates the use of a PKI service in a grid environment.

Authorization for Access Control

The authorization is a process to exercise access control of shared resources. Decisions can be made either at the access point of service or at a centralized place. Typically, the resource is a host that provides processors and storage for services deployed on it. Based on a set predefined policies or rules, the resource may enforce access for local services. The central authority is a special entity which is capable of issuing and revoking polices of access rights granted to remote accesses. The authority can be classified into three categories: attribute authorities, policy authorities, and identity authorities. Attribute authorities issue attribute assertions; policy authorities issue authorization policies; identity authorities issue certificates. The authorization server makes the final authorization decision.

Three Authorization Models

The subject is the user and the resource refers to the machine side. The subject-push model is shown at the top diagram. The user conducts handshake with the authority first and then with the resource site in a sequence. The resource-pulling model puts the resource in the middle. The user checks the resource first. Then the resource contacts its authority to verify the request, and the authority authorizes at step 3. Finally the resource accepts or rejects the request from the subject at step 4. The authorization agent model puts the authority in the middle. The subject check with the authority at step 1 and the authority makes decisions on the access of the requested resources. The authorization

process is complete at steps 3 and 4 in the reverse direction.

GRID SECURITY INFRASTRUCTURE (GSI)

Although the grid is increasingly deployed as a common approach to constructing dynamic, interdomain, distributed computing and data collaborations, —lack of security/trust between different services‖ is still an important challenge of the grid. The grid requires a security infrastructure with the following properties: easy to use; conforms with the VO's security needs while working well with site policies of each resource provider site; and provides appropriate authentication and encryption of all interactions.

The GSI is an important step toward satisfying these requirements. As a well-known security solution in the grid environment, GSI is a portion of the Globus Toolkit and provides fundamental security services needed to support grids, including supporting for message protection, authentication and delegation, and authorization. GSI enables secure authentication and communication over an open network, and permits mutual authentication across and among distributed sites with single sign-on capability. No centrally managed security system is required, and the grid maintains the integrity of its members' local policies. GSI supports both message-level security, which supports the WS-Security standard and the WS-SecureConversation specification to provide message protection for SOAP messages, and transport-level security, which means authentication via TLS with support for X.509 proxy certificates.

GSI Functional Layers

GT4 provides distinct WS and pre-WS authentication and authorization capabilities. Both build on the same base, namely the X.509 standard and entity certificates and proxy certificates, which are used to identify persistent entities such as users and servers and to support the temporary delegation of privileges to other entities, respectively. As shown , GSI may be thought of as being composed of four distinct functions: message protection, authentication, delegation, and authorization.

TLS (transport-level security) or WS-Security and WS-Secure Conversation (message-level) are used as message protection mechanisms in combination with SOAP. X.509 End Entity Certificates or Username and Password are used as authentication credentials.

X.509 Proxy Certificates and WS-Trust are used for delegation. An Authorization Framework allows for a variety of authorization schemes, including a —grid-mapfile‖

ACL, an ACL defined by a service, a custom authorization handler, and access to an authorization service via the SAML protocol. In addition, associated security tools provide for the storage of X.509 credentials (MyProxy and Delegation services), the mapping between GSI and other authentication mechanisms (e.g., KX509 and PKINIT for Kerberos, MyProxy for one-time passwords), and maintenance of information used for authorization (VOMS, GUMS, PERMIS).

The remainder of this section reviews both the GT implementations of each of these functions and the standards that are used in these implementations. The web services portions of GT4 use SOAP as their message protocol for communication. Message protection can be provided either by transport-level security, which transports SOAP messages over TLS, or by message-level security, which is signing and/or encrypting portions of the SOAP message using the WS-Security standard. Here we describe these two methods.

Transport-Level Security

Transport-level security entails SOAP messages conveyed over a network connection protected by TLS. TLS provides for both integrity protection and privacy (via encryption). Transport-level security is normally used in conjunction with X.509 credentials for authentication, but can also be used without such credentials to provide message protection without authentication, often referred to as —anonymous transport-level security.ll In this mode of operation, authentication may be done by username and password in a SOAP message.

Message-Level Security

GSI also provides message-level security for message protection for SOAP messages by implementing the WS-Security standard and the WS-Secure Conversation specification.

The WS-Security standard from OASIS defines a framework for applying security to individual SOAP messages; WS-Secure Conversation is a proposed standard from IBM and Microsoft that allows for an initial exchange of messages to establish a security context which can then be used to protect subsequent messages in a manner that requires less computational overhead (i.e., it allows the trade-off of initial overhead for setting up the session for lower overhead for messages).

GSI conforms to this standard. GSI uses these mechanisms to provide security on a per-message basis, that is, to an individual message without any preexisting context between the sender and receiver (outside of sharing some set of trust roots). GSI, as described further in the subsequent section on authentication, allows for both X.509 public key credentials and the combination of username and password for authentication; however, differences still exist. With username/password, only the WS-Security standard can be used to allow for authentication; that is, a receiver can verify the identity of the communication initiator.

GSI allows three additional protection mechanisms. The first is integrity protection, by which a

receiver can verify that messages were not altered in transit from the sender. The second is encryption, by which messages can be protected to provide confidentiality. The third is replay prevention, by which a receiver can verify that it has not received the same message previously. These protections are provided between WS-Security and WS-Secure Conversation. The former applies the keys associated with the sender and receiver's X.509 credentials. The X.509 credentials are used to establish a session key that is used to provide the message protection.

Authentication and Delegation

GSI has traditionally supported authentication and delegation through the use of X.509 certificates and public keys. As a new feature in GT4, GSI also supports authentication through plain usernames and passwords as a deployment option. We discuss both methods in this section. GSI uses X.509 certificates to identify persistent users and services.

As a central concept in GSI authentication, a certificate includes four primary pieces of information: (1) a subject name, which identifies the person or object that the certificate represents; (2) the public key belonging to the subject; (3) the identity of a CA that has signed the certificate to certify that the public key and the identity both belong to the subject; and (4) the digital signature of the named CA. X.509 provides each entity with a unique identifier (i.e., a distinguished name) and a method to assert that identifier to another party through the use of an asymmetric key pair bound to the identifier by the certificate.

The X.509 certificate used by GSI are conformant to the relevant standards and conventions. Grid deployments around the world have established their own CAs based on third-party software to issue the X.509 certificate for use with GSI and the Globus Toolkit. GSI also supports delegation and single sign-on through the use of standard X.509 proxy certificates. Proxy certificates allow bearers of X.509 to delegate their privileges temporarily to another entity. For the purposes of authentication and authorization, GSI treats certificates and proxy certificates equivalently. Authentication with X.509 credentials can be accomplished either via TLS, in the case of transport-level security, or via signature as specified by WS-Security, in the case of message-level security.

Trust Delegation

To reduce or even avoid the number of times the user must enter his passphrase when several grids are used or have agents (local or remote) requesting services on behalf of a user, GSI provides a delegation capability and a delegation service that provides an interface to allow clients to delegate (and renew) X.509 proxy certificates to a service. The interface to this service is based on the WS-Trust specification. A proxy consists of a new certificate and a private key. The key pair that is used for the proxy, that is, the public key embedded in the

certificate and the private key, may either be regenerated for each proxy or be obtained by other means. The new certificate contains the owner's identity, modified slightly to indicate that it is a proxy. The new certificate is signed by the owner, rather than a CA .

A sequence of trust delegations in which new certificates are signed by the owners rather by the CA. The certificate also includes a time notation after which the proxy should no longer be accepted by others. Proxies have limited lifetimes. Because the proxy isn't valid for very long, it doesn't have to stay quite as secure as the owner's private key, and thus it is possible to store the proxy's private key in a local storage system without being encrypted, as long as the permissions on the file prevent anyone else from looking at them easily. Once a proxy is created and stored, the user can use the proxy certificate and private key for mutual authentication without entering a password. When proxies are used, the mutual authentication process differs slightly. The remote party receives not only the proxy's certificate (signed by the owner), but also the owner's certificate. During mutual authentication, the owner's public key (obtained from her certificate) is used to validate the signature on the proxy certificate. The CA's public key is then used to validate the signature on the owner's certificate. This establishes a chain of trust from the CA to the last proxy through the successive owners of resources. The GSI uses WS-Security with textual usernames and passwords. This mechanism supports more rudimentary web service applications. When using usernames and passwords as opposed to X.509 credentials, the GSI provides authentication, but no advanced security features such as delegation, confidentiality, integrity, and replay prevention. However, one can use usernames and passwords with anonymous transport-level security such as unauthenticated TLS to ensure privacy.