Distributed System Security

Objectives

- Security Threats
- Security Policy
- Security Mechanisms
- Globus Security Architecture

Security threats

- way of looking at security in computer systems is that we attempt to protect the services and data it offers against security threats
- There are four types of security threats
  1. Interception
  2. Interruption
  3. Modification
  4. Fabrication

Interception

- The concept of interception refers to the situation that an unauthorized party has gained access to a service or data
  - Example
    - Where communication between two parties has been overheard by someone else
- Interception also happens when data are illegally copied
  - Example
    - after breaking into a person's private directory in a file system.

Interruption

- An example of interruption is when a file is corrupted or lost.
- More generally interruption refers to the situation in which services or data become unavailable, unusable, destroyed, and so on.
    - Example
        - denial of service attacks by which someone maliciously attempts to make a service inaccessible to other parties is a security threat that classifies as interruption

Modification

- involve unauthorized changing of data or tampering with a service so that it no longer adheres to its original specifications
- Example
  - Modifications include intercepting and subsequently changing transmitted data, tampering with database entries, and changing a program so that it secretly logs the activities of its user.

Fabrication

- Refers to the situation in which additional data or activity are generated that would normally not exist.
- Example
  - an intruder may attempt to add an entry into a password file or database. Likewise, it is sometimes possible to break into a system by replaying previously sent messages

- Note that interruption, modification, and fabrication can each be seen as a form of data falsification

Security Policy

- Simply stating that a system should be able to protect itself against all possible security threats is not the way to actually build a secure system.
- What is first needed is a description of security requirements, that is, a ***security policy***.

Security Policy(continues)

- A security policy describes precisely which actions the entities in a system are allowed to take and which ones are prohibited. Entities include users, services, data, machines, and so on.
- Once a security policy has been laid down, it becomes possible to concentrate on the ***security mechanisms*** by which a policy can be enforced.

Security Mechanisms

- Important security mechanisms are :
  1. Encryption
  2. Authentication
  3. Authorization
  4. Auditing

Encryption

- Encryption is fundamental to computer security
- Encryption transforms data into something an attacker cannot understand.
- In other words
  - encryption provides a means to implement data confidentiality.
- In addition, encryption allows us to check whether data have been modified.
- It thus also provides support for integrity checks.

Authentication

- is used to verify the claimed identity of a user, client, server, host, or other entity.
- In the case of clients, the basic premise is that before a service starts to perform any work on behalf of a client, the service must learn the client's identity (*unless the service is available to all*).
- Typically, users are authenticated by means of passwords, but there are many other ways to authenticate clients.

Authorization

- After a client has been authenticated, it is necessary to check whether that client is authorized to perform the action requested
- Example
  - Access to records in a medical database
    - Depending on who accesses the database. Permission may be granted to read records, to modify certain fields in a record, or to add or remove a record

Auditing

- Auditing tools are used to trace which clients accessed what, and which way.
- Although auditing does not really provide any protection against security threats.
- Audit logs can be extremely useful for the analysis of a security breach, and subsequently taking measures against intruders.

Auditing(continues)

- For this reason, attackers are generally keen not to leave any traces that could eventually lead to exposing their identity.
- In this sense, logging accesses makes attacking sometimes a riskier business.

Globus Security Policy

- To devise and properly use security mechanisms, it is necessary to understand what exactly needs to be protected, and what the assumptions are with respect to security.
- security policy for Globus entails eight statements

The environment consists of multiple administrative domains(#1)

- Globus assumes that the environment consists of multiple administrative domains, where each domain has its own local security policy.
- It is assumed that local policies cannot be changed just because the domain participates in Globus, nor can the overall policy of Globus override local security decisions.
- Consequently, security in Globus will restrict itself to operations that affect multiple domains

Local operations are subject to a local domain security policy only.(#2)

- operations that are initiated and carried out only within a single domain
- all security issues will be carried out using local security measures only.
- Globus will not impose additional measures

Global operations require the initiator to be known in each domain where the operation is carried out(#3)

- The Globus security policy states that requests for operations can be initiated either globally or locally.
- The initiator, be it a user or process acting on behalf of a user, must be locally known within each domain where that operation is carried out.

Operations between entities in different domains require mutual authentication (#4).

- An important policy statement is that operations between entities in different domains require mutual authentication.
- for example,
  - that if a user in one domain makes use of a service from another domain, then the identity of the user will have to be verified.

Global authentication replaces local authentication.(#5)

- If the identity of a user has been verified, and that user is also known locally in a domain, then he can act as being authenticated for that local domain.
- This means that Globus requires that its system wide authentication measures are sufficient to consider that a user has already been authenticated for a remote domain when accessing resources in that domain.
- Additional authentication by that domain should not be necessary

Controlling access to resources is subject to local security only (#6)

- Once a user has been authenticated, it is still necessary to verify the exact access rights with respect to resources.
- For example,
  - a user wanting to modify a file will first have to be authenticated, after which it can be checked whether or not that user is actually permitted to modify the file.

Users can delegate rights to processes. (#7)

- consider a mobile agent in Globus that carries out a task by initiating several operations in different domains, one after another. Such an agent may take a long time to complete its task.
- To avoid having to communicate with the user on whose behalf the agent is acting, Globus requires that processes can be delegated a subset of the user's rights.

A group of processes in the same domain can share credentials.(#8)

- Globus requires that groups of processes running with a single domain and acting on behalf of the same user may share a single set of credentials.
- credentials are needed for authentication.
- This statement essentially opens the road to scalable solutions for authentication by not demanding that each process carries its own unique set of credentials.

Globus Security Architecture

- Globus architecture is described using entities:
  - Users
  - User proxies: processes that are given permission to act on behalf of a user temporarily.
  - Resource proxies: processes used to translate a remote user’s requests into operations that do not violate a resource’s local security policy.
  - General processes
- The globus security architecture defines four different protocols,

user proxy and delegate rights to that proxy(#1)

- in order to let the user proxy act on behalf of its user, the user gives the proxy an appropriate set of credentials

Requesting the allocation of a resource in a remote domain(#2)

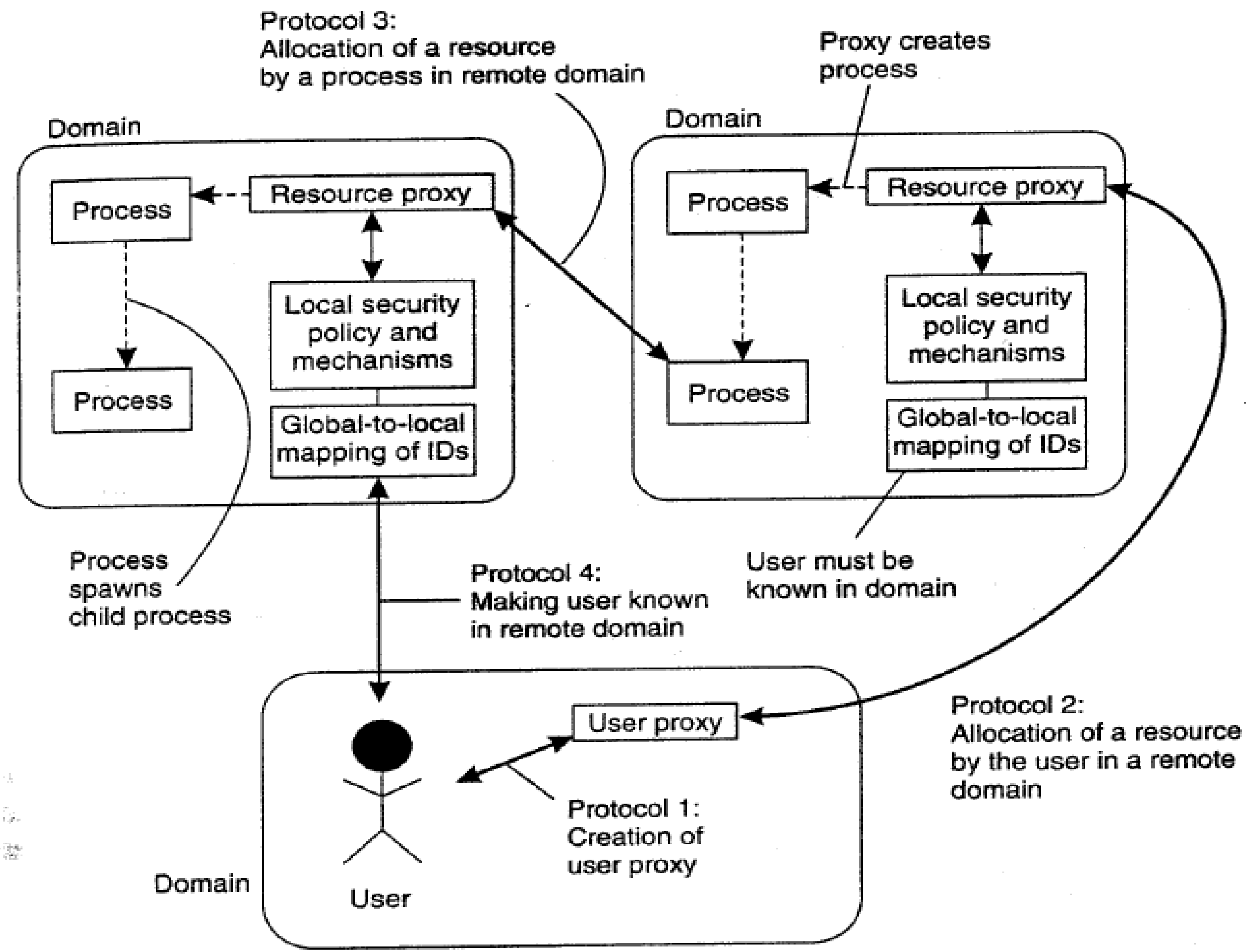
- the protocol tells a resource proxy to create a process in the remote domain after mutual authentication has taken place.
- That process represents the user, but operates in the same domain as the requested resource.
- The process is given access to the resource subject to the access control decisions local to that domain.

Allocating resources in a remote domain(#3)

- In the Globus system, this type of allocation is done via the user proxy, by letting a process have its associated user proxy request the allocation of resources, essentially following the second protocol.

Recognizing a User in a Remote Domain (#4)

- Assuming that a user has an account in a domain, what needs to be established is that the system wide credentials as held by a user proxy are automatically converted to credentials that are recognized by the specific domain.
- The protocol prescribes how the mapping between the global credentials and the local ones can be registered by the user in a mapping table local to that domain.



Protocol 3:
Allocation of a resource
by a process in remote domain

Proxy creates
process

Domain

Process

Resource proxy

Process

Local security
policy and
mechanisms

Global-to-local
mapping of IDs

Domain

Process

Resource proxy

Process

Local security
policy and
mechanisms

Global-to-local
mapping of IDs

Process
spawns
child process

Protocol 4:
Making user known
in remote domain

User must be
known in domain

User proxy

Protocol 2:
Allocation of a resource
by the user in a remote
domain

Protocol 1:
Creation of
user proxy

Domain

User

References

- Andrew S.Tanenbaum & Maarten Van Steen. Distributed Systems – Principles and Paradigms. 2nd ed. 2007.