10.2 Controlled Process Creator

This pattern addresses how to define and grant appropriate access rights for a new process.

Example

Most operating systems create a process with the same rights as its parent. If a hacker can trick an operating system into creating a child of the supervisor process, this runs with all the rights of the supervisor.

Context

An operating system in which processes or threads need to be created according to application needs.

Problem

A user executes an application composed of several concurrent processes. Processes are usually created through system calls to the operating system [Sil03]. A process that needs to create a new process gets the operating system to create a child process that is given access to some resources. A computing system uses many processes or threads. Processes need to be created according to application needs, and the operating system itself is composed of processes. If processes are not controlled, they can interfere with each other and access data illegally. Their rights for resources should be carefully defined according to appropriate policies, for example ‘need-to-know.’

The solution to this problem must resolve the following forces:

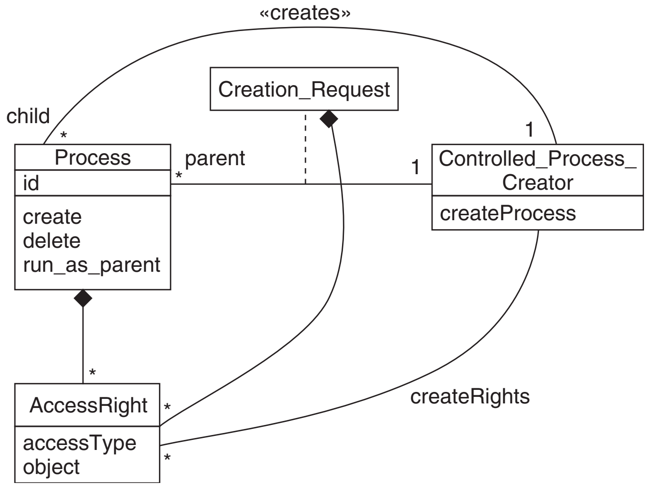
- There should be a convenient way to select a policy to define process’ rights. Defining rights without a policy brings contradictory and non-systematic access restrictions that can be easily circumvented.
- A child process may need to impersonate its parent in specific actions, but this should be carefully controlled, otherwise a compromised child could leak information or destroy data.
- The number of child processes created by a process must be restricted, or process spawning could be user to carry out denial-of-service attacks.
- There are situations in which a process needs to act with more than its normal rights, for example to access data in a file to which it doesn’t normally have access.

Solution

Because new processes are created through system calls or messages to the operating system, we have a chance to control the rights given to a new process. Typically, operating systems create a new process as a child process. We let the parent assign a specific set of rights to its children, which is more secure because a more precise control of rights is possible.

Structure

The figure below shows the class diagram for this pattern. The `Controlled Process Creator` is a part of the operating system in charge of creating processes. The `Creation Request` contains the access rights that the parent defines for the created child. These access rights must be a subset of the parent's access rights.



Class diagram for CONTROLLED PROCESS CREATOR

Dynamics

The figure on page 331 shows the dynamics of process creation. A process requests the creation of a new process. The access rights passed in the creation request is used to create the new access rights for the new process.

Implementation

For each required application of kernel threads, define their rights according to their intended function.

Example Resolved

There is now no automatic inheritance of rights in the creation of children processes, so creating a child process confers no advantage for a hacker.

Known Uses

In some hardened operating systems such as Hewlett Packard's Virtual Vault, a new set of rights must be defined for each child [HP].

Consequences

The following benefits may be expected from applying this pattern:

- The created process can receive rights according to required security policies.
- The number of children produced by a process can be controlled. This is useful to control denial of service attacks.
- The rights may include the parent's id, allowing the child to run with the rights of its parent.

The following potential liability may arise from applying this pattern:

- Explicit rights transfer takes more time than using a default transfer.

See Also

CONTROLLED EXECUTION ENVIRONMENT (346) could use this pattern to define the execution domain of new processes.