CSE 509 Lecture 3

Prof. Rob Johnson, Scribe:Arun Rathakrishnan

September 10, 2013

1 Review

1.1 Standard as a trusted entity

A trusted entity can also be an implementation of a standard or even a standard by itself. It is possible that both the entities can violate trust. For example, the inclusion of Null encryption in protocols, may be viewed as an attempt to violate trust.

1.2 Protocol Downgrade Attack

In this attack, one side of communication is forced to lower the level of encryption, making it easier to violate trust and attack the system.

For example, consider a banking site that supports https and http protocols. If your browser allows you to connect using http through an unsecure network, it may be subject to a man-in-the-middle attack. The attacker can talk to the victim using http, while using in formation from that communication to talk in https to the bank. The bank sends https response to the attacker, which can be converted on the fly to http and presented to the victim.

Here the attacker successfully lowers the level of encryption in his communication with the victim. Modern browsers can be forced to use SSL to establish https connections with a website during the first time, the browser visits the site. This can help to prevent this attack.

2 Access Control

Access control is a mechanism by which a service provide is able to provided a trusted service to a client it does not trust. The service provider,

- receives request from a client.
- Needs to identify which clients can perform which actions.
- Needs to implement the decisions.

The below are three access control mechanisms.

- Access Control Matrix (Role Based Access Control)
- Access Control Lists
- Bell, Lapadula/ Biba Model

We will consider a kernel as a service provider which needs to service requests to its file system. The common operations allowed are read and write for files; lookup, create, link, unlink (remove), etc for directories. The kernel may also support super-block operations.

3 Access Control Matrix

The access control matrices control - who (subject) can do what (operation/permission) to what (object). The subject is typically a user (in a multi-user OS), or an application (in an Android kernel). We consider the Harrison Rizzo Ullman model, where each row in the matrix corresponds to a subject - a single user. User groups are not considered. Each column corresponds to an object and each cell to the operation allowed for a subject on a particular object.You can add subjects and columns to the matrix.

The access control matrix, irrespective of its representation, is an object whose access needs to be controlled too. So it has an entry as to who can change the operations for a given object, identifying the subject as an owner. In unix file system, a file's owner can change the operations for otherusers (we do not consider groups here).

In addition to the matrix, the model contains rules which can be used to modify the matrix. The rules are typically listed in `Pre-Condition: Operation` format. A rule is executed if the precondition (based on the access control matrix data) is satisfied, which results in permission bits of a user being changed for a file.

3.1 Example rule for chmod

```
chmod(executor, object, user, permission):
    if(owner is in A[executor][object]):
        A[user][object] <- permission
```

3.2 Example rule for chown

Only a root user can change the ownership of a file in unix filesystems.

```
chown(executor, object, newOwner):
        if(executor == owner):
             Remove the owner permission from previous owner of object
             A[newOwner][object] <- owner
```

3.3 HRU Theorem

Given an initial state of the access control matrix and a list of rules for updating the matrix and a final state of the access control matrix, to determine whether the application of rules on the list takes the matrix from initial state to final state is undecidable.

As a result, we can never figure out if a known, but undesirable security violation can be averted by using an access control matrix through a sequence of updates. Typically, most subjects do not have access to most objects and hence, the matrix is too sparse to be used in practice. Access Control Lists are more practical. They store for each object, only subjects with valid permissions and the access rights. Roughly, a column of access control matrix corresponds to Access Control List for that object.

4 Monotonicity/Order Dependence

We assume that groups can contain a subset of users, and can appear as subjects in access control lists. The permission for an object for any subject now may consider user permissions and a combination of several groups' permission to which a user can belong.

Some of the policies can be order dependent like,

- first match policy
- last match policy

Some of them can be order independent like,

- most specific/ most general
- most permissive/ most restricted

4.1 Most specific policy in Unix

Unix file systems check for a file if,the user is an owner or has user permissions in ACL or belongs to a group which is listed as a subject in ACL or if not, finally considers the other permission bits.

4.2 Most permissive policy in Unix

Assume Unix has two lists as a substitute for ACL. The first list is a list of users for files. The second is the list of groups for a file. It can search through all the valid subjects, (user's own entry or any of user's group in ACL for that file) and select the most permissive policy for that file.

4.3 Negative Entries

A permission bit may be a negative entry, which expressly prevents an operation on an object. IP Tables, Windows Files System, Firewalls use such entries in their ACL.

Most restrictive or most specific policy rules may be enforced strictly. It is also possible to expressly prevent some operations on objects by a subject without using negative entries.

5 Bell/ Lapadula

Objects and subjects are classified based on a level and a set called compartment. Levels are usually tagged as,

- Top Secret
- Secret
- For Official Use Only (FOUO)
- Public

A compartments form subsets of information classification that further restrict access within a level. Compartments apply across all levels.

An object labelled with level and compartment (L, CS) can be accessed by a subject with level and compartment (L', CS') only if, $L \leq L'$ and $CS \subset CS'$.

6 Mandatory Access Control Systems

An access control system is discretionary if the subjects are able to determine or change the access level of objects. In a mandatory access control system, no user can determine or change the access level of any object. For example, you may combine a Top Secret document and a Secret document to form a new document with level Secret in a discretionary access control. But in mandatory access control, the system will determine the level to be Top Secret, which can not be changed by subjects. Such a policy is called a "No write down" policy.