



**A tutorial report for SENG 609.22**

**Agent Based Software Engineering**

**Course Instructor: Dr. Behrouz H. Far**

## **Tutorial on Security for Web-Based Applications**

**Submitted By: Rahul Mehrotra**

### **1. Introduction**

The social and economic impact of World Wide Web is mind boggling but the fact is that Internet is here to stay. It has empowered us to conduct online shopping, talking, dating and information sharing in every segment of our society. Businesses can share and exchange information for more efficient business practices. While extremely useful for conducting day-to-day business operations, the proliferation of e-commerce over the Internet has provided a perfect target for computer crackers, script-kiddies and other such bad elements. The reliance of a business on the Internet makes it extremely vulnerable to all sorts of attacks.

Completely securing a computer against unauthorized access is extremely difficult. There are many ways for an attacker to gain access. In general however, an attacker employs the easiest way to fulfill his or her malicious intentions. Some of these attacks include

shoulder surfing, dumpster diving, network sniffing, exploiting code weaknesses, denial of service attacks and others. These attacks can come from outside as well as from within. Hence, it is equally important to provide adequate safeguards for both internal and external threat sources. . With evolving technologies, enabling new economic models via increasingly integrated and distributed business environments; security has an even higher priority.

## 2. Basic Terminology

Security is multidimensional concept. Some of these dimensions include privacy, physical access restrictions, application availability, network confidentiality, content integrity and access policy. Security is all about managing risks. When people think of security, they generally refer to one or more of the following aspects.

- *Authentication*: The process of verifying an identity claimed by or for a system entity.
- *Access Control*: Protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy and is permitted by only authorized entities according to that policy.
- *Audit Trail*: A chronological record of system activities to enable the examination of the sequence of activities surrounding an operation in a security-relevant transaction from inception to final results.
- *Confidentiality*: The property that information is not made available or disclosed to unauthorized individual, entities or processes.
- *Integrity*: The property that information has not been modified or destroyed in an unauthorized manner.
- *Availability*: The property of a system or a system resource being accessible and usable upon demand by an authorized system entity.
- *Nonrepudiation*: A security service that provides protection against false denial of involvement in a communication.

## 3. Security Issues for Web-Based Applications

Typically, a Web-based application can be represented as a multi layered architecture depicted in figure 1 below. It includes a Web client, network servers, and a backend info-

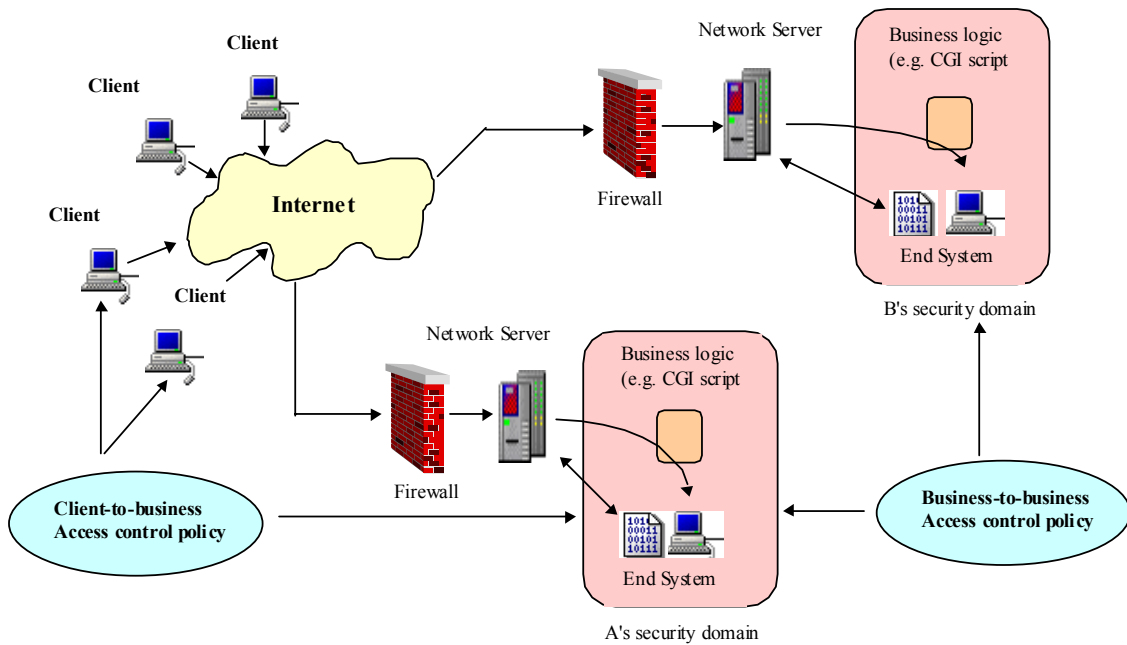


Figure 1: Multi-layered architecture for Web-based applications

formation systems supported by a suite of databases. For transaction-oriented applications, such as e-commerce, middle-ware is usually provided between the network servers and backend systems to ensure proper interoperability.

Considerable security challenges and vulnerabilities exist within each component of this architecture. End users are exposed to several security and privacy risks when using Web browsers. Information about a user such as login name or machine name can be collected and used to profile the user, thus raising serious privacy concerns. Cookies, the data stored on the client's machine and exchanged between the Web client and Web Server to maintain connection information, can be used for the purpose of gathering such information. A source of vulnerability at the client site also comes from the use of executable content on the Web, such as Java Applets, ActiveX controls, and the like. Network servers are the places where most network services are located, such as the Web Server, the mail server, and so forth. Firewall technology has become the most popular defense for these servers against the open untrusted Internet. Though firewalls can prevent illegitimate traffic from traveling from the Internet to corporate networks, legitimate requests that pass through a firewall may be used for data-driven attack on the networks or backend systems.

Existing public key infrastructures (PKIs) provide encryption mechanisms for ensuring information confidentiality, as well as digital signature techniques for authentication, data integrity, and non-repudiation. As no access authorization services are provided in this approach, it has a rather limited scope for Web-based applications. The fact that insider attacks constitute a considerable threat further emphasizes the need for robust host-based security, whereby substantial authentication and access control services must be deployed at the host. The web primarily uses a hypertext approach for information dissemination. With the growth of e-commerce applications, the Web is rapidly being transformed into

an activity- or transaction-intensive environment. Security models for hypertext-based systems are not many. For the Web, access models and mechanisms should facilitate dynamic changes in the content and context of information, allow monitoring of the state of the system, and facilitate carrying out transactional activities. Existing access models gradually need to evolve these features.

#### 4. Access Control Models

Several models have been proposed to address the access control requirements of distributed applications. The following sections present the salient points of various models and also an assessment of their suitability for supporting Web based applications.

##### Discretionary Access Control (DAC) Model

In DAC models, all the subjects and objects in a system are enumerated and the access authorization rules for each subject and object in the system are specified. Subjects can be users, groups, or processes that act on behalf of other subjects. If a subject is the owner of an object, the subject is authorized to grant or revoke access rights on the object to other subjects at his discretion. DAC policies are flexible and the most widely used for Web-based applications.

Among the existing representations of DAC models, a noticeable one is the access control matrix (ACM) model. An access control matrix has users represented on the rows and protected objects on the columns (Table 1). The entries in the matrix describe what type of access each user has to each object.

Table 1: Access Control Matrix for DAC model

Users \ Objects	KIMS FILE	DONS FILE	Payrol 1	Payrol 2	DOES FILE
Kim	rw	r	rw	r	
Joe		r			
Don		rw	r		
Jones			r		
Doe				rw	
Mgr Jim	cp	cp	c	c	c
Jan			rw	rw	

Access type "r" denotes read access

Access type "w" denotes write accesses

Access type "c" means control permission

Access type "cp" means control with passing ability

The basic principle of discretionary access control contains a fundamental flaw that makes it vulnerable to Trojan horses. For example, DAC allows copying of data from one object to another, which can result in allowing access to a copy of data to a user who does not have access to the original data. Such risks can propagate to the entire Web environment, causing serious violation of security goals.

## **Mandatory Access Control (DAC) Model**

In a MAC model, all subjects and objects are classified based on predefined sensitivity levels that are used in the access decision process. An important goal of a MAC model is to control information flow in order to ensure confidentiality and integrity of the information. For example, to ensure information confidentiality in defense applications a MAC model can be implemented using a multi-level security mechanism with labels such as TOP SECRET, SECRET, and CONFIDENTIAL that uses no read-up and no write down rules, also known as Bell-LePadula restrictions. These rules are designed to ensure that information does not flow from a higher sensitivity level to a lower sensitivity level. To achieve information integrity, the access rules are formulated as no read-down and no write up. The goal in this case is not to allow the flow of low integrity information to high integrity objects. Unlike DAC, "Mandatory" means that the system enforces the policy; users do not have the discretion to share their files.

MAC models provide more robust protection mechanisms for data, and deal with more specific security requirements, such as an information flow control policy. However, enforcement of MAC policies is often a difficult task, and in particular for Web-based applications, they do not provide viable solutions because they lack adequate flexibility. Originally, MAC and DAC models were not intended for Web-based applications. In particular their design philosophy was not intended to serve to serve hypertext-based systems. The hypertext information model uses special objects such as links, frames, document nodes, and so forth all of which need to be protected.

## **Role-based Access Control (RBAC) Model**

RBAC is a form of mandatory access control, but not based on multilevel security requirements. Rather, access control decisions are determined by the roles individual users take on as part of an organization. RBAC tends to be modeled after the natural structure of an organization where functions are grouped into roles and users are permitted to one or more of these roles. The security policy of the organization determines role membership and the allocation of each role's capabilities. The RBAC models have been shown to be "policy neutral" in the sense that using role hierarchies and constraints, a wide range of security policies can be expressed including traditional DAC and MAC. Security administration is also greatly simplified by the use of roles to organize access privileges. For example, if a user moves to a new function within the organization, the user can simply be assigned to the new role and removed from the old one, whereas in the absence of an RBAC model, the user's old privileges would have to be individually revoked, and new privileges would have to be granted. Special administrative roles can be designated to manage other roles. Unlike DAC, RBAC's premise is that the organization, not the user, owns the objects being secured. In most implementations, users cannot sub-delegate access permissions on to other users at their discretion.

Using an RBAC model is a highly desirable goal for addressing the key security requirements of Web-based applications in general and Workflow Management Systems

(WFMSs) in particular. Roles can be assigned to work flow tasks so that a user with any of the roles related to a task may be authorized to execute it. A key feature of RBAC is its potential support for a multi-domain environment, which makes it an attractive candidate for Web-based applications. Role-hierarchy mapping between two RBAC-based policy domains can be used to define a metapolicy for secure interoperation.

### **Access Control Models for Tasks and Workflow**

The models discussed above use the subject-object view toward security. These models have a limited scope and are not flexible enough to allow access policies based on the content of information or the nature of the tasks/transactions in a WFMS. WFMSs have emerged as a key technology for enabling transaction-intensive Web applications. A viable approach is to use RBAC framework to enforce security requirements during the execution of workflow tasks but substantial extensions are needed to address security issues related to Web applications and WFMSs. Researchers have proposed a family of task-based access control (TBAC) models that constitutes four models arranged in the form of a hierarchy. The TBAC0 model represents the base model that provides the basic or the minimum facilities such as tasks, authorization steps and their dependencies. The TBAC1 model is an extension of TBAC0 that includes the composite authorizations of two or more authorization steps. The TBAC2 model is another extension of TBAC0 that allows both static and dynamic constraints. The TBAC3 model is a consolidated model that has features of both the TBAC1 and TBAC2 models.

### **Agent-based Approach**

With the increase of Internet applications, software agents are becoming popular as an emerging system building paradigm. This paradigm can be effectively used to provide security features for Web applications. An agent is a process characterized by adaptation, cooperation, autonomy, and mobility. Some agent communications languages can be used to negotiate policies during conflicts for secure interoperation among participating policy domains. Agents can be assigned to security enforcement tasks at the servers and client machines. Although mobility and adaptability are essential to the efficient use of Internet resources, they pose several security threats. For example, an agent can engage in malicious behavior, thus disrupting normal operation of the host. Similarly, a host may be able to affect the activity of an agent by denying required access to local information resources.

### **Certificate-based Approach**

Public-key infrastructure technology is maturing, and the use of PKI certificates is expected to be ubiquitous in the near future. Certificates issued by a PKI facility can be used for enforcing access control in the Web environment. An example is the use of an extended X.509 certificate that carries role information about a user. A certification authority that acts as trust center in the global environment issues these certificates. The use of public-key certificates is suitable for simple applications. These techniques can be used to either support a host's access control method by carrying access control information or provide a separate access control mechanism based on trust centers.

## 5. Conclusion

Achieving secure interoperation in a heterogeneous Web environment is a difficult task, because of the inherent dynamism. Using RBAC models and software security agents are suitable approaches for such environments. The RBAC models have several desirable features such as flexibility, policy neutrality, better support for security management and administration, the principle of least privilege, and other aspects that make them attractive candidates for developing secure Web-based applications. Furthermore, an RBAC model provides a natural mechanism for addressing the security issues related to the execution of tasks and workflow. A key advantage of RBAC models is the ease of their deployment over the Internet. The use of RBAC in conjunction with PKI facilities can provide a pragmatic approach to addressing issues related to security of Web-based applications and WFSMs.

The intent of this tutorial is not to provide a how-to-manual or an all-encompassing definition of security. The attempt here is to highlight the current and future dimensions of security. The comparative assessment of existing security models in terms of supporting Web-based applications are presented here to know more about security threat facing software industry and Web-base applications in particular. Several extensions to the existing models are needed to develop the workable solutions to adequately address the security threats.