SecTag: A Multi-Policy Supported Secure Web Tag Framework

Ruixuan Li, Meng Dong, Bin Liu, Jianfeng Lu, Xiaopu Ma, Kai Li
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, Hubei 430074, China
rxli@hust.edu.cn, {mengdong, bliu9, lujianfeng, xpma}@smail.hust.edu.cn, likai@hust.edu.cn

INTRODUCTION

Web-based enterprise-level applications have gained a tremendous growth in the last decade. Traditional access control module development in web systems often suffers problems, such as lack of fine-grained and multi-policy support, tight coupling of the access control logic with the business logic. It is hard to reconfigure or modify the access control policies after the system has been deployed. Hence, it is ultimately important to provide a development framework considering the features of multi-policy authorization, component reusability, multi-views and high interaction support. As shown in Figure 1, web services are different according to access permissions.

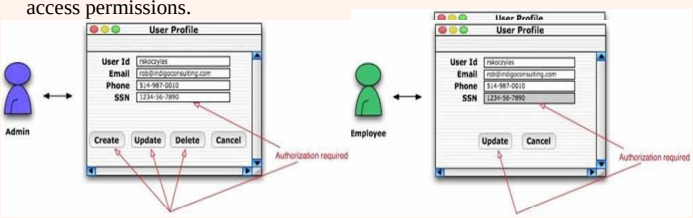

Fig.1 Web services are different according to access permissions.

DESIGN

FRAMEWORK

Our goal is to design and implement a light-weight basic framework of SecTag, which can be easily satisfied with the traditional model-view-controller (MVC) mode. By using SecTag, fine-grained access control policies can be easily configured through the visual interface without any modification of codes. Most of the general access control logics can be encapsulated in the form of secure tags. Thus, the development and maintenance workload are greatly reduced. Figure 2 shows the architecture of SecTag framework.

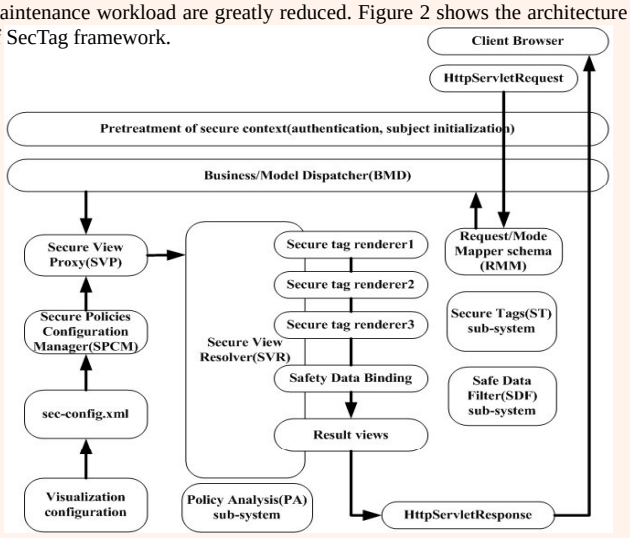

Fig.2 Framework of SecTag.

POLICY CONFIGURATION

Based on actual secure applications, tag rendering that responds to user's requests can be divided into three states: normal, view-only and unavailable, taking submit tag as an example shown in Figure 3.

•Tags of user interface (UI) display: mainly includes the visibility and availability of control.

•Tags of data access control: mainly includes the dynamic rendering of the data list.

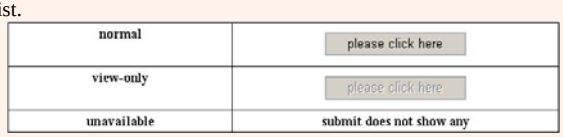

Fig.3 Three rendering states of submit tag.

The structure of policy configuration is shown in Figure 4. Each policy can set attribute "mode" and "access", and contains at least one specific rule. "rule" is used to describe the specific access control information, and its attribute "access" specifies the access permission to the rule. we can inject a structured query language (SQL) statement or a method of a class in sec-config.xml when "type" is defined as "SQL" or "Method". In these cases, Policy Analysis Subsystem (PAS) will automatically load the dynamic data to complete the policy analysis.

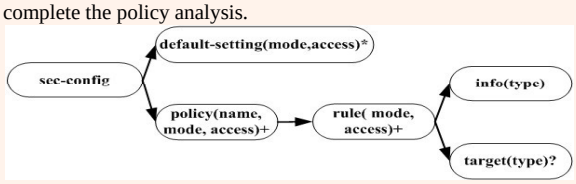

Fig.4 Architecture of Secure Tag Subsystem.

SECURE TAGS

We extend traditional web tags to support secure attributes through binding secure policies that are configured in secconfig.xml. We use FreeMarker template engine to dynamically render tags, which makes different users see different views according to their requests and permissions. The architecture of Secure Tag Subsystem (STS) is shown in Figure 5. These secure tags are based on the development guidelines of JSP 2.0. Secure tags in SecTag includes five main components.

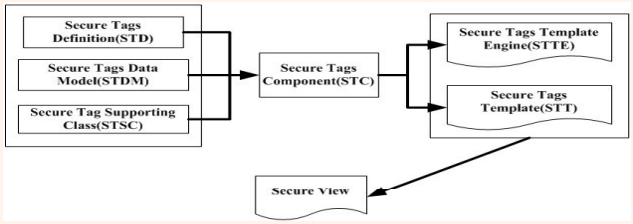

Fig.5 Structure of policy configuration.

EXAMPLE

Taking the tag select that receives data as an example, we bind attribute "policy" of select to policy p that is configured in sec-config.xml shown in Figure 6, and describe the user identity in the form of (username, role, level). Suppose there are 6 user profiles: (Tom, SuperAdmin, L6), (Mary, Admin, L5), (Lucy, User, L4), (Lily, User, L3), (John, Casualuser, L2), (Mark, Casualuser, L1). The rendering result views are shown in Figure 7 (a), (b) and (c) with the login of Tom, Lucy and John respectively. All the data items could be selected for Tom, but view-only for Lucy. John can select items 1~4 normally, view-only for items 5~8 and unavailable for the rest items.

```
<sec-config>
  <default-setting access="view-only" mode="RBAC"/>
  <policy name="p">
    <rule access="view-only" mode="MAC">
      <info>L3</info>
    </rule>
    <rule access="unavailable" mode="DAC">
      <info>John,Mark</info>
      <target>n,n,n,n,v,v,v,v</target>
    </rule>
    <rule access="normal" mode="RBAC">
      <info type="SQL">SELECT user FROM UserRole
        WHERE role="SuperAdmin" OR role="Admin"</info>
      <target type="Method">
        cn.edu.hust.idc.sectag.demo.getTargetList()</target>
    </rule>
  </policy>
</sec-config>
```

Fig.6 An example of policy p.

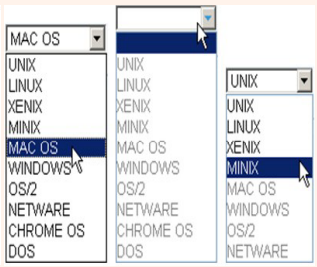

Fig.7 Rendering of secure tag select.

CONTRIBUTIONS

We develop a reusable secure development framework SecTag to solve the problem of tight coupling between access control and business logic. It assists developers to achieve fine-grained and multi-policy authorization management for web resource protection.

FOR FURTHER INFORMATION

Please contact *rxli@hust.edu.cn* or visit *http://idc.hust.edu.cn*. Source code can be downloaded from *http://code.google.com/p/sectag/*.