A Single Sign-on Scheme Supporting Multiple Authentication Modes Based on TNC

Wu Kun
Department of Computer
Beijing University of Posts and Telecommunications
Beijing, China
sherrywukun@163.com

Bai Zhongying
Department of Computer
Beijing University of Posts and Telecommunications
Beijing, China
sherrywukun@msn.com

Abstract—The existing Single Sign-On (SSO) schemes are based on PKI/PMI, WS-Security, Kerberos etc. But they can't support Trusted Network Connect (TNC). To make up for the shortage, a multi-authentication modes SSO scheme based on TNC named SSO-TNC is proposed. SSO-TNC uses an authentication gateway to realize SSO by loading different authentication plugins, issuing Cookies and binding the original accounts of applications. Furthermore, SSO-TNC combines Federated TNC (FTNC) to implement the SSO across domains based on security certificate. The experiment results show that SSO-TNC can effectively solve the SSO problem in the TNC environment, and provide better security, availability and scalability.

Keywords—Trusted network connect, Single sign-on, Multiple authentication modes

I. INTRODUCTION

SSO is an identity authentication method that authenticates a user only once to access multiple applications by managing users’ accounts uniformly in an authorized security context [1, 2]. SSO simplifies effectively the work flow and maintenance, and improves the security and performance of the whole system.

The TNC Work Group defines an open solution architecture that enables network operators to enforce policies regarding endpoint integrity when granting access to a network infrastructure [3]. In the network access layer, the Network Access Requestor (NAR) establishes network access; the Network Access Authority (NAA) verifies the NAR’s identity and integrity to authorize the NAR; the Network Access Enforcer (NAE) enforces the NAA’s decision. So, SSO is implemented at the network access layer.

However, it doesn’t refer to the SSO in the TNC environment. In order to accomplish unique identity authentication, it is necessary to research the SSO in trusted platform. Also, it plays an important role in extending trusted computing mechanism and perfecting access control mechanism.

Under the framework of TNC, this paper comes up with a new SSO scheme supporting multiple authentication modes named SSO-TNC. SSO-TNC makes use of an authentication gateway to shield the implementation detail of every authentication mode, which only needs to add a related plugin to submit the relevant identity. And, the authentication gateway realizes SSO by binding the original accounts of applications. A client communicates with the authentication gateway and accesses resources of applications through a portal.

We have researched a FTNC model based security certificate called Security Certificate Federated Access Model (SCFAM) [4]. After the Security Posture Information Collector (SPIC) has verified an endpoint’s identity and integrity, the Security Certificate Authority (SCA) issues a Security Certificate (SC) to the endpoint. The endpoint submits the SC to access a Service Provider (SP). Thus, SSO-TNC combines SCFAM to realize the SSO across domains.

The experiment results show SSO-TNC can effectively solve the SSO problem based on TNC, and provide better security, availability and scalability.

The remainder of this paper is structured as follows: Section 2 points out the SSO-TNC architecture and its work flow. The experiment results are given in section 3. Summaries and References are given in section 4 and 5.

II. DESIGN OF SSO-TNC

A. Design of SSO

As illustrated in Fig.1, the SSO-TNC architecture incorporates five parts: the Client, the Portal, the Authentication Gateway (AG), the Authentication Server (AS) and the Application Server (ApS).

![SSO-TNC architecture](image)

Figure 1. SSO-TNC architecture

In order to get resources of the ApSs, the Client requests to verify its identity by different authentication modes. It corresponds to the NAR in TNC. The Portal integrates applications in local domain. The AG integrates different authentication plugins to communicate with the related ASs. The Portal and the AG together correspond to the NAE in TNC, and that the AS corresponds to the NAA. In addition, the AG binds the original accounts of applications in local domain and maintains the binding information. In the
meantime, the AG can locate the security domain of the Client.

We take the PKI/PMI mode as an example to introduce the work flow of SSO-TNC as illustrated in Fig. 2.

---

**Figure 2. Work flow of SSO-TNC**

1. **Step 1:** A Client requests resources of the ApSs by the PKI/PMI mode.
2. **Step 2:** The Client is redirected to the AG.
3. **Step 3:** The AG calls the PKI/PMI authentication plugin, and requires the UsbKey. The Client submits the PKCc.
4. **Step 4:** The AG reads the IDInfo and sends it to the AS. The AS queries and verifies the PKCc and the related PMI attribute certificate from the LDAP server. The AS generates the loginCookie and the grantCookie, and stores the loginCookie. Then, the AS sends the grantCookie to the AG.
5. **Step 5:** The AG stores the grantCookie and sends it to the Client.
6. **Step 6:** The Client sends the grantCookie to access the Portal again.
7. **Step 7:** The Portal verifies the grantCookie from the AG. If invalid, the AG will deny access. If valid, the AG will find the binding account of application and permit access.
8. **Step 8:** The Portal returns the result to the Client.

---

**Figure 3. Work flow of cross-domain SSO**

---

**B. Design of cross-Domain SSO**

When a Client accesses resources in another security domain, the AS can’t authenticate its identity and grant access, which leads to the problem of cross-domain SSO. To solve the problem, this paper uses security certificate in SCFAM to create the relationship of cross-authentication for the AGs in different security domains. Fig. 3 gives the work flow of cross-domain SSO.

---

**III. RESULT AND ANALYSIS OF EXPERIMENT**

There are three experiments to prove the validity, security and performance of SSO-TNC respectively.

Experiment 1: Test the username/password mode 10 times, the digital certificate mode 10 times, the fingerprint mode 10 times. They all succeed.
Experiment 2: Test the anti-attack ability for 10 groups (10 times per group). The statistical result shows that the anti-attack success ratio of every group is above 70%.

Experiment 3: By coding with multiple threads, simulate 150 virtual users accessing concurrently and 700 virtual users accessing online. Table 2 shows the result data, which shows that response is faster and CPU and RAM of the AS is utilized reasonably.

Note: The Response Time column indicates the average response time of the system.

The CPU column indicates the CPU average utilization ratio of the AS.

The RAM column indicates the RAM average utilization ratio of the AS.

IV. SUMMARIES

SSO-TNC in this paper uses the AG to support multiple authentication modes, and to bind the original accounts of applications. The authentication details are shielded. The plugins and portal provide better availability and scalability. At the same time, SSO-TNC combines the SCFAM model to realize the SSO across domains. A series of anti-attack methods ensure the system security. The experiment results turn out that SSO-TNC is valid, secure and reliable. The future work may designate the priority for different authentication modes based on the role of a user, and accomplish access control based on different grains and constraint conditions. Moreover, we will add the system auditing and monitoring.

REFERENCES


<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Client</td>
<td>IDInfo</td>
<td>Client’s identity information</td>
</tr>
<tr>
<td>P</td>
<td>Portal</td>
<td>PKCc</td>
<td>Client’s public key certificate</td>
</tr>
<tr>
<td>A</td>
<td>AG</td>
<td>Kc, Sc</td>
<td>Client’s public key and private key</td>
</tr>
<tr>
<td>S</td>
<td>PKI/PMI AS</td>
<td>Ka, Sa</td>
<td>AG’s public key and private key</td>
</tr>
<tr>
<td>Mode1</td>
<td>PKI/PMI mode</td>
<td>Ks, Ss</td>
<td>AS’s public key and private key</td>
</tr>
<tr>
<td>loginCookie</td>
<td>login ticket</td>
<td>N1, N3</td>
<td>AG’s one-off random numbers</td>
</tr>
<tr>
<td>grantCookie</td>
<td>grant ticket</td>
<td>N2, N4</td>
<td>Client’s one-off random numbers</td>
</tr>
<tr>
<td>Ts</td>
<td>AS’s timestamp</td>
<td>validity</td>
<td>Cookies’ validity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Runtime</th>
<th>Response Time (sec)</th>
<th>Success (times)</th>
<th>Failure (times)</th>
<th>Success Ratio (%)</th>
<th>CPU Ratio (%)</th>
<th>RAM Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>00:10:19</td>
<td>1.549</td>
<td>7317</td>
<td>1</td>
<td>99.986</td>
<td>12.4</td>
<td>18.4</td>
</tr>
<tr>
<td>700</td>
<td>00:27:53</td>
<td>2.968</td>
<td>130315</td>
<td>551</td>
<td>99.6</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>