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 authorizated 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).

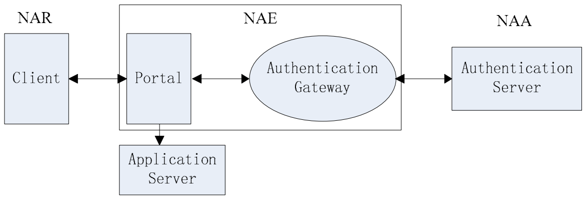

Figure 1. SSO-TNC architecture

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

Published by Atlantis Press, Paris, France.

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.

1	C → P	{C, Mode1}Sc
2	P → A	{C, Mode1}Sc
3	C ← A	{{C, A, N1}Sa}Kc
	C → A	{{C, A, PKCc, N1, N2}Sc}Ka
4	A → S	{{S, A, IDInfo, N3}Sa}Ks
	S	loginCookie={S, C, Ts, validity}Ks
		grantCookie={{S, C, Ts, validity}Ss}Ka
	A ← S	{{A, S, grantCookie, N3}Ss}Ka
5	C ← A	{{C, A, grantCookie, N2}Sa}Kc
6	C → P	{{A, C, grantCookie, N4}Sc}Ka
7	P → A	{{A, C, grantCookie, N4}Sc}Ka
	P ← A	page, {{C, A, page, N4}Sa}Kc
8	C ← P	{{C, A, page, N4}Sa}Kc

Figure 2. Work flow of SSO-TNC

Step 1: A Client requests resources of the ApSs by the PKI/PMI mode.

Step 2: The Client is redirected to the AG.

Step 3: The AG calls the PKI/PMI authentication plugin, and requires the UsbKey. The Client submits the PKCc.

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.

Step 5: The AG stores the grantCookie and sends it to the Client.

Step 6: The Client sends the grantCookie to access the Portal again.

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.

Step 8: The Portal returns the result to the Client.

When the Client accesses every application, the AG generates the temporary sessionCookie that indicates the accessed state of application.

The symbols define as illustrated in Table 1.

SSO-TNC utilizes Cookies to pass the authentication message, which support not only current popular browsers but also different authentication modes. The validity of loginCookie is 8 hours or until logout. The validity of grantCookie is 1 hour or until quiting access. The validity of sessionCookie is 10 minutes or until closing the browser.

In consideration of common attacks (e.g. message tamper, Man in the Middle (MitM) attack), there makes use of the secret-key distribution and management system to encrypt and sign the important information and messages. In the meanwhile, the timestamp can prevent replay attack. In the transport layer, the TLS protocol is used to encrypt and encapsulate.

When the Client logs off, it requests to the Portal. The Portal sends the request to the AG. The AG deletes all the Cookies of the Client and sends the request to the AS. The AS deletes the loginCookie of the Client and sends the success message to the AG. The AG sends the message to the Portal. The Portal informs the Client.

B. Design of cross-Domain SSO

When a Client accesses resources in anothoer 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.

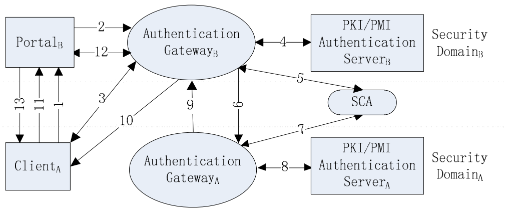
Portal$_B$
2
12
Authentication Gateway$_B$
4
PKI/PMI Authentication Server$_B$
Security Domain$_B$
5
SCA
13
11
1
3
10
9
6
7
Client$_A$
Authentication Gateway$_A$
8
PKI/PMI Authentication Server$_A$
Security Domain$_A$

Figure 3. Work flow of cross-domain SSO

Step 1: ClientA in DomainA requests resources of the ApSs in DomainB by the PKI/PMI mode.

Step 2: ClientA is redirected to AGB by PortalB.

Step 3: AGB requires ClientA`s UsbKey.

Step 4: AGB reads the IDInfo and sends it to ASB. Because ASB hasn`t found the PKCc from the LDAP server, it returns the failure message to AGB.

Step 5: AGB locates the security domain about ClientA. If AGB has not a SC, it will register and apply for a SC from the SCA. If has, it will go on.

Step 6: AGB sends the SC to AGA.

Step 7: AGA verifies the SC from the SCA. If invalid, ClientA will be denied. If valid, it will go on.

Step 8: AGA authenticates ClientA`s identity by communicating with ASA. ASA generates and stores the loginCookie and the cross-loginCookie, then sends the cross-loginCookie to AGA.

Step 9: AGA stores the cross-loginCookie and sends it to AGB.

Step 10: AGB stores the cross-loginCookie, then generates the grantCookie and sends it to ClientA.

Step 11: ClientA sends the grantCookie to access PortalB.

Step 12: PortalB verifies the grantCookie from AGB. If invalid, AGB will deny access. If valid, AGB will find the binding account of application and permit access.

Step 13: PortalB returns the result to ClientA.

When ClientA accesses every application, AGB generates the sessionCookie.

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.

Published by Atlantis Press, Paris, France.

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

[1] Lin Manshan; Guo Heqing. Actuality and Development of Single Sign-On Technology. Computer Applications [J], 2004, PP:248-250 (in Chinese).

[2] Li Xiaobiao; Wen Qiaoyan; Dai Zhanfeng. A Supporting Multi-Mode Application Single Sign-On Scheme Based on PKI/PMI. Journal of Beijing University of Posts and Telecommunications [J], 2009, PP:104-108 (in Chinese).

[3] TCG Trusted Network Connect. TNC Architecture for Interoperability Specification Version 1.4 Revision 4. http://www.trustedcomputinggroup.org/files/resource_files/51F9691E-1D09-3519-AD1C1E27D285F03B/TNC_Architecture_v1_4_r4.pdf [EB/OL], 2009.

[4] BAI Zhongying; WU Kun. Research of Federated Trusted Network Connect Based on Security Certificate. Proceedings of 2011 3rd International Conference on Computer and Network Technology [C], 2011, PP:621-625.

TABLE I. SYMBOLS OF AUTHENTICATION COMMUNICATION

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

TABLE II. PERFORMANCE TESTING RESULTS

Type	Runtime	Response Time (sec)	Success (times)	Failure (times)	Success Ratio (%)	CPU Ratio (%)	RAM Ratio (%)
150	00:10:19	1.549	7317	1	99.986	12.4	18.4
700	00:27:53	2.968	130315	551	99.6	-	-

Published by Atlantis Press, Paris, France.