Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

Single Sign on for Secure Authentication of Web Services using Kerberos

Pranav Bhagwat
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Vinayak Tapkir
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Kiran Andure
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Komal Sorte
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

***Abstract*— Single Sign-on (SSO) makes it possible for users to only log on once and then access different services via automatic authentication by using the same credential. However, majority of existing SSO schemes do not satisfy security notions or require a high trust level on a trusted third party (TTP), even though SSO has become popular in new distributed systems and computer networks. Motivated by this fact, we formalize a new security scheme of single sign-on, which satisfies strong security notions but also has a low trust level on TTP. We propose to design an easy to use, safe, efficient solution of Web Single Sign-On, referring to the design idea of Kerberos protocol based on ticket access and using Single Sign-On model. We then propose a generic construction of SSO using digital signatures, and present concrete initialization. The proposed SSO scheme is secure according to our new formal infrastructure. We note that our descriptive study investigates the link between SSO and Kerberos using digital signatures which also be of an independent interest.**

Keywords*— Single Sign-on , Trusted third party, Kerberos, Security scheme.

I. INTRODUCTION

Due to massive growth of internet the users have to maintain multiple accounts for which separate credentials are to be used which are required to be memorized making the task a little cumbersome. Meanwhile it also increases the burden on service providers as they have to perform the credential management like issuing, updating, revocation for which they have to exhaust additional resources like keeping an extra server just to handle the password reset requests. Single sign-on (SSO) is an authentication mechanism that allows users to only log on one time and then access different services via automatic authentication by using the same single credentials. This means that our SSO model may be an effective way to make users relax at their daily work. We propose to design a Single Sign On system which allows the users to log on once and then access different services via automatic authentication and same credentials can be used for logging into multiple registered accounts. Our system design promises an easy to use, safe, efficient solution of Web Single Sign-On, referring to the design idea of Kerberos protocol based on ticket access and using Single Sign-On model. Although there are several researches focusing on single sign on, but some have flaws should be limitations. Therefore our task is to formalize a generalized and optimized model of Single Sign on to overcome those flaws.

II. RELATED WORK

In 1995, Parker [8] defined the various Technologies and The Products of Single Sign-on Systems. Scripting and Access Tickets are the two main approaches of single sign-on. In 2000, Lee and Chang proposed [4] a user identification and key distribution scheme to maintain user anonymity in distributed computer networks. Later, Wu and Hsu pointed out that Lee-Chang scheme is insecure against both impersonation attack and identity disclosure attack. Meanwhile, Yang et al. identified a weakness in Wu-Hsu scheme and proposed an improvement. In 2004, Gang et al [2] proposed two designs of Single Sign-On and discuss its advantages and disadvantage of these two versions. In 2007, Suriadi et al [11] find out that there is a lack of built-in privacy mechanisms within the current identity management systems. They give a proposal for the extension of existing Federated Single Sign-On (FSSO) systems to adopt the beneficial properties of the User-Centric Identity Management (UCIM) model to provide an identity management system that allows the users to control and enforce their privacy requirements while still retaining the convenient features of FSSO. In 2007, Maryam [6] et al demonstrates a centralized password-based authentication system using SSO for Web based application in distributed environments. Centralized, Distributed and Federated Approaches are introduced and Cookie capabilities are used for implementation of this system that is called centralized cookie-based SSO or CC-SSO. In 2008, David [1] proposed a framework for single sign-on by using an EMV card for two-factor authentication. Single sign-on by using an EMV card does not need the card making physical contact with the network connected device and without exposing the keys and PIN that are used to protect financial transactions. The proposed method could improve the protection for the card, the cardholder and the service provider(s). In 2009, Magyari et al [5] proposed a single sign-on mechanism which is based on certificates generated on request for client applications. In 2009, Rajesh and Alwyn [9] discussed that several Single Sign-On frameworks were proposed and implemented so far but they are not balance in Security, Efficiency and Usability. Their proposed framework using Identity Based Encryption System (IBES) instead of Public key infrastructure (PKI). In 2010, Moo Nam Ko et al [7] discussed Facebook Connect services which allow users to login to other websites using their Facebook identity and information and which will then potentially feed back to a users Facebook network information about their actions on the site.

Facebook Platform allows users to import their identity, profile, privacy policy, social graph and content from Facebook to third-party sites. OAuth is an open standard for authorization, commonly used as a way for Internet users to log into third party websites using their Microsoft, Google, Facebook or Twitter accounts without exposing their password. Generally, OAuth provides to clients a 'secure delegated access' to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server. OAuth began in November 2006 when Blaine Cook was developing the Twitter OpenID implementation. OAuth 2.0 is the next evolution of the OAuth protocol and is not backwards compatible with OAuth 1.0. OAuth 2.0 doesn't support signature, encryption, channel binding, or client verification.

OAuth 2.0 has had numerous security flaws exposed in implementations. The protocol itself has been described as inherently insecure by security experts and a primary contributor to the specification stated that implementation mistakes are almost inevitable.

III. KERBEROS BASICS AND AUTHENTICATION SCHEME

3.1 Definition

Kerberos is a computer network authentication rotocol which works on the concept of 'ticket granting' without actual transmission of free text passwords over the insecure internet allowing the nodes to communicate with each other in a secure manner.

Kerberos is built on symmetric key cryptography requiring a trusted third party as well as public key cryptography during certain phases of authentication.

Kerberos can be broken down as-

- a protocol for authentication
- uses tickets to authenticate
- avoids storing passwords locally or sending them over the internet
- involves a trusted 3rd-party
- built on symmetric-key cryptography

3.2 Kerberos domain

Admins creates domains called Kerberos realms that will encompass all that is available to access. Here it is granted that all the users may not have access to certain services or host machines that is defined within the policy management also called as access control policies. A realm defines what Kerberos manages in terms of who can access what.

Your machine i.e. the Client, lives within this realm, as well as the service that you want to request and the Key Distribution Center, KDC.

3.3 Protocol description

Kerberos can be described using three terms viz. Authentication Server, Ticket Granting Server, Trusted Third Party. The client logs itself into the Authentication Server (AS) and a username request is sent to a key distribution center (KDC). The KDC issues an encrypted ticket-granting ticket (TGT) using the user's password and returns the encrypted result to the user's workstation. Every TGT has a Time-to-live also called as expiry time and is the time until which that TGT can be used for an active user session.

The client sends the TGT to the ticket-granting service (TGS) to communicate with other nodes, which usually shares the same host as the KDC. If the TGT is valid, the user is permitted to access the requested service, the TGS issues a ticket and session keys, which are returned to the client. The client then sends the ticket to the service server (SS) along with its service request.

3.4 Interactions and integration

When requesting access to a service or host, three interactions take place-

- Client and the Authentication Server.
- Client and the Ticket Granting Server.
- Client and the Service or host machine.

With each interaction client will receive two messages. Each message is one that you can decrypt, and one that you can not.The service or machine you are requesting access to never communicates directly with the KDC. The KDC stores all of the secret keys for user machines and services in its database. Secret keys are passwords plus a salt that are hashed – the hash algorithm is chosen during implementation of the Kerberos setup. For services or host machines, there are no passwords (who would enter it). A key is actually generated by an admin during initial setup and memorized on the service/host machine. Again, these secret keys are all stored in the KDC database; recall the Kerberos reliance on symmetric-key cryptography. The KDC itself is encrypted with a master key to add a layer of difficulty from stealing keys from the database. There are Kerberos configurations and implementations that use public-key cryptography instead of symmetrical key encryption.

IV. SSO SYSTEM COMPONENTS AND ARCHITECTURE

4.1. Trusted Third party(TTP)

4.1.1. Web Application Server.

Web Application Server is a server computer that will deliver Web pages. The Web server has an IP address and possibly a domain name. The responsibility of the web server is to fetch and process the webpages requested by the user and send it to the browser.

4.1.1.2. Mysql Server.

Mysql server is the term used to refer to the back-end system of a database application. It will perform tasks such as data storage, data manipulation, archiving, and other non-user specific tasks.

4.2. Key Distribution Center

4.2.1. Authentication Server.

The authentication server working in co-operation with the local database performs the job of validating the user authenticity like Active Directory Server in windows domain

4.2.2 Ticket Granting Server.

It performs the operation of providing Service Ticket after getting TGT(Ticket Granting Ticket) from TTP.

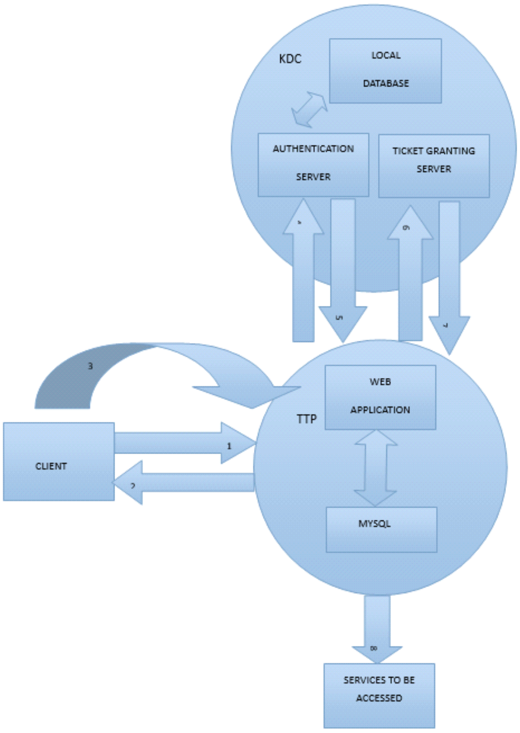

Figure 1 Proposed single sign-on system

V. CONCLUSION

Here we have made a sincere descriptive study to build an optimized Single Sign On system using Kerberos authentication which satisfies the standard security notions which allows users to log in once and securely access multiple SSO enabled applications resulting into administrative complexity reduction and greater user satisfaction.

ACKNOWLEDGEMENT

We are highly indebted to our project guide Prof. A. M. Bhadgale for their continuous support, supervision, motivation and guidance throughout the tenure of our project. We deeply express our gratitude to our project mentor Mr. K. V. Kokil working as a scientist at ARDE, Pune(Armament Research and Development Establishment) without whom the project could not be completed.

REFERENCES

[1] David J. Boyd (2008), " Single Sign-On to the Web with an EMV Card", International Symposium on Collaborative Technologies and Systems, 2008.CTS2008, DOI: 10.1109/CTS.2008.4543920, pp(s) 112-120.

[2] Gang Zhao, Dong Zhengand, Kefei Chen (2004), " Design of Single Sign-On", Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business, DOI 10.1109/CEC.EAST.2004.34, page(s):253-256.

[3] Gupta V.K., Sheetlani Jitendra, Gupta Dhiraj and Shukla Brahma Datta, “Concurrency Control and Security issues of Distributed Databases Transaction”, Research Journal of Engineering Sciences, NIMS University, Jaipur, Rajasthan, INDIA, Vol. 1(2), 70-73, August (2012).

[4] Lee W. B. and Chang C. C., “User identification and key distribution maintaining anonymity for distributed computer networks”, Computer Systems Science and Engineering, Page (s): 113-116, 2000.

[5] Magyari A., Genge B., Haller P. (2009)," Certificate-Based Single Sign-on Mechanism for Multi-Platform Distributed Systems", Electrical and Mechanical Engineering, Page(s): 113-123, 2009.

[6]] Maryam Eslami Chalandar, Parviz Darvish and Amir Masoud Rahmani (2007) “A Centralized Cookie-Based Single Sign-On in Distributed Systems”, Information and Communications Technology, ICICT 2007, ITI 5th International Conference on Dec 2007.

[7] Moo Nam Ko, Cheek, G.P., Shehab, M., Sandhu, R., "Social-Networks Connect Services", IEEE Computer Society on August 2010, Page(s): 37-43, ISSN: 0018-9162, DOI: 10.1109/MC.2010.239.

[8] Parker T.A. (1995), "Single Sign-on Systems - The Technologies And The Products", European Convention on Security and Detection', 16-18 May 1995, Conference publication NO. 408, pp. 151-155. © IEEE.

[9] Rajesh and Alwyn R. (2009), "Secure Web Based Single Sign-On (SSO) framework using Identity Based Encryption System", International Conference on Advances in Recent Technologies in Communication and Computing, DOI 10.1109/ARTCom.2009.82, pp.430-432 © 2009 IEEE.

[10] "Security Issues", available on http://docs.oracle.com/cd/B12037_01/network.101/b10777/overview.htm

[11] Suriadi Suriadi, Ernest Foo and Audun Jøsang (2007), "A User-centric Federated Single Sign-on System", IFIP International Conference on Network and Parallel Computing, 2007, pp. 99-106.

[12] T. Sivasakthi and Dr. N Prabakaran , “Applying Digital Signature with Encryption Algorithm of User Authentication for Data Security in Cloud Computing”,International Journal of Innovative Research in Computer and Communication Engineering. Vol. 2, Issue 2, February 2014.

[13] Jingquan Wang, Guilin Wang, and Willy Susilo, “Secure Single Sign-on Schemes Constructed from Nominative Signatures”, 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.