[Baravade*, 5(1): January, 2016]

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A REIVEW ON EXTENSIVE SECURITY OF COMMERCIAL WEB SSO SYSTEMS

Mr.D.S. Baravade *,Mr.Y.R. Gurav,Mr.D. R. Kale

* Computer Engineering Department , Ashokrao Mane Group Of Institution,India

ABSTRACT

The research of this paper are focused on extensive security analysis of commercial web SSO systems. Single sign-on solutions are a safe and great alternative to credential loss. In this paper we observe advantages and challenges that come with implementing single sign-on .In addition to this, we are discovering and confirming new flaws in other web SSO systems. Goal of this paper are focused on improving and to expose security weakness in services using Web Single sign On (SSO) secure mechanism, In this paper we provide comparative review of existing work done on Web SSO,Like there categaries,implementation issue and challenges of Web SSO. Next we discuss mechanisms in which SSO is carried out to provide well Security.

KEYWORDS: Authentication, Web Service, Logic Flaw, Secure Protocol, Single-Sign-On,

INTRODUCTION

Single sign-on (SSO) is a latest authentication mechanism that enables a web user with a single credential to be authenticated by many service providers in a distributed network. Imagine that you visit amgoi.com, a leading shopping website, try to get in your accounts there. Here is what you will see amgoi.com allows you to sign in using your Facebook,gmail account, This way of authentication is known as single sign-on (SSO), which enables a user to log in once and gain access to multiple websites without the repeatedly typing her/his passwords. Today, leading web technology companies such as Google ,Facebook, , Yahoo, Twitter all offer SSO services. Such services, which we call web SSO, Our study shows that not only do logic flaws pervasively exist in web SSO deployments, but they are practically discoverable by the adversary through analysis of the SSO steps disclosed from the browser, even though source code of these systems is unavailable. The web SSO systems we found to be vulnerable include those of Facebook, Google ID, PayPal Access, Freelancer etc. We reported our findings to related parties and helped them fix those bugs, for which we were acknowledged in various ways, e.g., public recognitions.

SSO Categories

*Password synchronization, Enterprise SSO, Federation, "True" SSO, Web SSO[3].*Password synchronization as an "old school" approach where multiple systems have unique usernames but a common password. Enterprise SSO is flexible and allows users to preserve the password complexity rules for individual applications, but this approach can be often more difficult to implement and typically leaves small pockets of non-coverage due to integration[11] difficulties or a choice to not extend the technology to certain systems for expense or integration purposes.

Web SSO is similar to the enterprise approach, he says, but with a web front-end. Peterson describes True SSO as "any technology that gives a one username, one password, (and) one credential type of experience. This is what Microsoft built for the Windows universe with Active Directory, and now technologies exist that can extend that experience to applications and systems that are able to consume the AD credential." Federation is a type of Web SSO that uses standards such as SAML or Windows Federated Authentication to create a trusted relationship between two unrelated organizations or systems[3]. In a federated relationship, he says, there is a service provider that provides the token or credential or permission to log on and an identity provider that serves up the identity of the one logging in. Using the federation approach, the receiving system or organization trusts the person logging in

from the originating system or organization and that the user is approved to ask for the access in question. It is SSO because usually the token can be re-used across numerous federated applications,.“The bigger and more complex an organization is, the less likely it is to be able to get away with one flavor of SSO," he adds. "Most (companies) choose a blended approach that includes some or all of the types listed above, or they choose to implement SSO where it matters most and selectively ignore certain systems from an SSO standpoint."

Implementing SSO

While implementing SSO ,Four recommendations for companies planning an SSO implementation:

1)We notice that SSO is a journey, not a destination -- you actually are never done as new applications come online or new access methods are introduced

2)Give priority to which systems and/or user populations matter most and then choose the right SSO approach for those.

3)Force what already exists in the company, as much as possible.

4)If the company decides to employ alliance, look at the bigger picture and select an SSO method that doesn't restrict the implementation to a small pocket of systems or users.

Some author says SSO is popular because it improves productivity and reduces the possibility that employees will use a lot of easy-to-crack passwords. Eliminating separate passwords for each application means that users will be better able to remember one more complex password than lots of passwords likely to be less complex.One important consideration when building out an SSO infrastructure is determining which department will own the responsibility for the technology. Often the debate comes down to the IT team, which is operations-focused and dedicated to making sure users have easy access to their applications, versus the security team, which is risk-averse and focuses on protecting data more so than ease of use.

SSO Deployment Questions

If your company is planning a new or expanded SSO deployment,Some authors recommends that you consider these questions to assist in determining how to get started:

1. Who will be managing the SSO system and what is their level of familiarity and expertise with the technology?
2. Will the SSO purchase grow as needs evolve?
3. Is it a new SSO deployment or an upgrade?
4. What are the main pain points the company is trying to solve? Centralizing Access? Authorization? Auditing? If all three, what are their specific needs around each?
5. What is the level of complexity involved? How many target systems need to be integrated? Are there any applications that need to be managed by the SSO system that are not owned by the company (i.e., a bank needing access to a partner's trading application)?

When SSO is used, it is important that new, appropriate security levels for critical data are added from day one so that the most important data is always the most secure.

Challenges with SSO

One of the biggest challenges to making SSO work efficiently is that data that identifies the user is not always consistent throughout the enterprise. For example, user John might be identified as John for one application, but J. Smith for another and Smith, John for a third. While it is possible for some systems to normalize the different names based on John Smith's title and job description, there is a possibility that there are more than one person with that

name in the company with the same title, such as a sales representative.A global profile needs to include additional attributes of the individual so that one user can be distinguished from another. Such attributes could include location of their office, the name of the person to whom they report, authorizations for access to certain types of data, and unique personnel ID numbers.There will always be a small percentage of results that cannot be determined using the basic logic tree approach.Ideally the company using the SSO software will develop audit and access control policies that will normalize as many of the different ways to identify users as possible. However, for those instances when the company cannot normalize the differences, such as when the company is dealing with a cloud-based application where data is stored on a system outside the control of the corporate IT department, policies should be in place to identify and flag possible conflicts.

RELATED ANALYSIS

Web Single Sign-On: a View from the Browser

SSO is essentially a process for an IdP to convince an RP that because this browser has signed onto the IdP as Alice, this same browser is now granted the capability to sign onto the RP as Alice. The tricky part here is that the IdP must bind Alice's capability to the correct browser that truly represents Alice. In all existing SSO systems, such a binding is through proof-by-possession: Alice's browser needs to present to the RP a token issued by the IdP to demonstrate that it possesses the capability that the IdP grants to Alice. Security of an SSO scheme depends on how the token is handled, so the browser naturally undertakes many critical steps, and thus is the focus of our investigation.

Browser relayed message (BRM).

An SSO process can be described as a sequence of browser relayed messages exchanged between the to website, an HTTP communication can be thought of as a sequence of request-

Studying SSO schemes on major websites

Like a debugger extracting ground truths about call stack, memory and registers, the BRM analyzer extracts necessary ground truths about an SSO scheme to be studied, e.g., what Bob could read or write, especially some key elements (e.g., those labeled with SEC or SIG, etc.) . With this tool, we now can go onto the field study about leading commercial web SSO systems. The study covers popular SSO services on the web (e.g., Facebook, Google, JanRain and PayPal), and the SSO systems of high-profile websites/services (e.g., FarmVille, Freelancer, Nasdaq and Sears). The result shows that these prominent web SSO systems contain serious logic flaws that make it completely realistic for an unauthorized party to log into their customers' accounts. These flaws are also found to be diverse, distributed across the code of RPs and IdPs, and at the stages of login and account linking. We elaborate these vulnerabilities in the rest of the section.

Understanding the SSO vulnerabilities

All the logic flaws described in the paper, no matter how subtle they are, were all discovered through a simple and rather mechanical procedure at the high level:Understand whether the SSO is based on a secret token or an authentic token. Accordingly, there are only two types of problems – authentic token forged by Bob,or either a secret token sent to Bob. Locate the token in BRMs. Understand how it is propagated or how it is covered by a signature. Apply adversary scenarios to BRMs, which corresponds to the only three strategies – Bob acting as another client, Bob acting as another RP and Bob acting as a page in Alice's client. Today's web SSO systems often fail to fully understand the security implications during token exchange, particularly, how to ensure that the token is well secured and correctly verified, and what the adversary is capable of doing in the process.Variations in the vulnerabilities. With this complexity, we feel that it can be hard to speculate about how a system can go wrong before looking at its details. This is why a lot of detailed investigations need to be conducted with human analyst's creativity and domain knowledge. We do believe, however, that for known vulnerabilities, one can build a tool to

automatically identify other websites suffering from similar problems, but it is not the focus of this paper.RP developers' due diligence. The variations are in the non-trivial details of individual systems. In this study, we spent a great focuson demonstrating such variations. This variety comes from the way SSO services are integrated: each RP can integrate the same SSO service differently; the security of the integration depends not only on the program logic on RP and IdP sites, but also on the underlying web platform. The difficulty in implementation and system details suggest that it can be hard for IdP developers to anticipate all possible RP implementations in the world. Because RP developers are the people who put together a concrete system, they are naturally the final gatekeeper for its security. We accept that most RP developers today may not realize the necessity of such a due attentiveness, but merely consider SSO implementation as a task of calling individual APIs on IdPs.We believe that an analysis like what we did is helpful,. Developers are obviously in a better position to conduct the analysis than us, as they know precisely which data serve as the primary user ID, the underlying system features that the RP code relies on, and other insider knowledge.

CONCLUSION

In this paper, We represents SSO Catagaries ,their implementation issues and challenges faced in SSO .we also analys an extensive security study of commercial web SSO systems. The study shows that security- critical logic flaws pervasively exist in these systems. We involved our analysis steps performed on commercial SSO systems and how they lead to discovery. Every discovered flaw allows the attacker to sign in as the victim.

In addition to those reported, we are discovering and confirming new flaws in other web SSO systems. We reached the end of this work the overall work can be summed up with the following statements. The task consisted first of researching and investigating within the around protocol with the focus on its aspect of single sign on mechanism. Also this work proposes further research into more efficient enhancements for security of single sign on for distributed computer networks. For third-party sites, credential generation and synced, cloud-based storage can be provided.

REFERENCES

[1] Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, Llanos Abad. "Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps," ACM FMSE, 2008
[2] Devdatta Akhawe, Adam Barth, Peifung Lam, John Mitchell, Dawn Song. "Towards a Formal Foundation of Web Security," IEEE Computer Security Foundations Symposium, 2010
[3] Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, G. Pellegrino, A. Sorniotti. "From Multiple Credentials to Browser-based Single Sign-On: Are We More Secure?" IFIP Information Security Conference (SEC), 2011
[4] Karthikeyan Bhargavan, Cédric Fournet, Andrew D. Gordon, Nikhil Swamy. "Verified implementations of the information card federated identity-management protocol, ACM ASIACCS 2008.
[5] Blue Research. "Consumer Perceptions of Online Registration and Social Sign-In," http://janrain.com/consumer-research-social-signin
[6] Andrew Bortz, Adam Barth, and Alexei Czeskis. "Origin Cookies: Session Integrity for Web Applications," W2SP 2011.
[7] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. ACM Trans. Computer
[8] Weidong Cui, Jayanthkumar Kannan, Helen J. Wang. "Discoverer: Automatic Protocol Reverse Engineering from Network Traces," USENIX Security Symposium 2007
[9] Facebook. "White hats," http://www.facebook.com/whitehat
[10] Facebook. "OAuth Dialog," http://developers.facebook.com/docs /reference/dialogs/oauth/
[11] Facebook Developers. "Legacy Canvas Auth," http://developers.facebook.com/docs/authentication/fb_sig/
[12] Fiddler Web Debugger. http://www.fiddler2.com/fiddler2
[13] Google Code. "Federated Login for Google Account Users," http://code.google.com/apis/accounts/docs/OpenID.html

[14] Thomas Groß. "Security analysis of the SAML single sign-on browser/artifact profile," ACSAC 2003
[15] S. M. Hansen, J. Skriver, and H. R. Nielson. "Using static analysis to validate the SAML single sign-on protocol," Workshop on Issues in the Theory of Security, 2005
[16] Brian Kissel. "OpenID 2009 Year in Review," http://openid.net/2009/12/16/openid-2009-year-in-review/
[17] LocalConnection (in flash.net). http://help.adobe.com/en_US /FlashPlatform/reference/actionscript/3/flash/net/LocalConnection.ht ml?filter_flex=4.1&filter_flashplayer=10.1&filter_air=2
[18] Los Angeles Times. "The Sims Social bests FarmVille as the second-largest Facebook game," http://latimesblogs.latimes.com/ entertainmentnewsbuzz/2011/09/sims-social-surpasses-farmville-as-second-largest-facebook-game.html
[19] Catherine Meadows. "Language Generation and Verification in the NRL Protocol Analyzer," Computer Security Foundations 1996.
[20] Microsoft. "INFO: Internet Explorer Does Not Send Referer Header in Unsecured Situations," http://support.microsoft.com/kb/178066
[21] Jonathan K. Millen. "The Interrogator Model," IEEE Symposium on Security and Privacy 1995.
[22] OASIS Standard. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.
[23] OpenID Wiki. "OpenID Phishing Brainstorm," http://wiki. openid.net/w/page/12995216/OpenID_Phishing_Brainstorm
[24] Birgit Pfitzmann and Michael Waidner. "Analysis of Liberty Single-Sign-on with Enabled Clients," IEEE Internet Computing, 7(6) 2003.
[25] San-Tsai Sun, Eric Pospisil, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, Konstantin Beznosov. "What Makes Users Refuse Web Single Sign-On? An Empirical Investigation of OpenID," Symposium On Usable Privacy and Security, 2011