بررسی مساله احراز هویت کاربران به کمک مکانیزم تک ورود

شکوفه بستان[۱]
[۱] دانشگاه آزاد اسلامی، واحد علوم و تحقیقات، گروه کامپیوتر، یزد، ایران، Shekoofe.bostan@yahoo.com
محمدرضا ملاخلیلی میبدی
دانشگاه آزاد اسلامی، واحد میبد، گروه کامپیوتر، میبد، ایران، mollakhalili@maybodiau.ac.ir

چکیده- آنچه مسلم است بشر همیشه برای کوتاه‌تر کردن مسیرها و سهولت بخشیدن به امور در تلاش است، یکی از مسائلی که امروزه ذهن اکثر کاربران را به خود درگیرکرده، تنوع نام کاربری و کلمات عبوری است که برای ورود به هر قسمت از سیستم باید به خاطر بسپارند، همچنین مدیریت این کلمات عبور و تعیین نوع دسترسی‌ها از سوی مدیر، کار دشواری است اما با استفاده از مکانیزم تک ورود یا *single sign on* می‌توان تنها با یک بار ورود به سیستم و استفاده از یک رمز عبور به تمام آن منابع و امکانات دست یافت.
در این مقاله، به معرفی مکانیزم تک ورود یا *SSO* پرداخته و ساختار و معماری آن را بررسی خواهیم کرد. همچنین انواع آن را از نظر نحوه به کارگیری در محیط‌های مختلف بیان کرده و به معرفی پروتکل‌های آن می‌پردازیم.
کلید واژه- اعتبارنامه، امنیت، تک ورود، کربروس، منابع.

۱- مقدمه

در بیشتر سازمان‌ها با گذر زمان، نرم افزارهای مختلفی ایجاد می‌شوند که دارای روش‌های احراز هویت مختص خودشان هستند و با ادامه این روند، افراد با معضل تعدد نام کاربری و کلمه‌های عبور مواجه می‌شوند که به خاطر سپردن تمام آن‌ها کار دشواری است. بنابراین نیاز به وجود سیستمی که به کمک آن تنها با یک بار احراز هویت، کاربر قادر به دسترسی به تمام منابع و اطلاعات مرتبط با وی که از نظر دسترسی دارای مجوز یکسانی است باشد، احساس می شود[8]. در واقع سیستمی برای کاهش تعداد دفعاتی که کاربر بایستی اطلاعات کاربری خود را به هنگام استفاده از منابع چند گانه وارد کند لازم است (شکل ۱). سیستمی که بتواند چنین نیازی را پاسخ دهد، اصطلاحا تک ورود یا SSO نامیده می شود.
روش‌های مختلفی برای پیاده سازیSSO مطرح می‌شود مانند پروتکل کربروس[a]، گواهینامه X.509 و پروتکل SAML [5].
همچنین می‌توان از SSO در سیستم‌های مختلف و با ساختارهای متفاوتی از شبکه های محلی تا شبکه جهانی استفاده کرد.
مزایای به کارگیری SSO عبارت است از :

- افزایش سودمندی به دلیل رضایت بیشتر کاربران با کاهش تعداد دفعات ورود به سیستم و تعداد دفعات فراخوانی روالهای بازیافت اطلاعات کاربری
- کاهش مشکلات مدیر شبکه/سیستم در مدیریت حجم وسیع اطلاعات کاربری
- افزایش امنیت سیستم
- کاهش هزینه
- صرفه جویی در زمان
- قابلیت کنترل دسترسی کاربران
- دسترسی سریع و آسان
- قابلیت اطمینان

اما بزرگترین ایرادی که بر آن وارد است به دلیل داشتن تنها یک رمز عبور اصلی است که در صورت گم شدن آن رمز، کلیه برنامه‌های سیستم با خطر جدی مواجهند. همچنین مکانیزم احراز هویت[b] کاربر در سیستم از اهمیت زیادی برخوردار است چرا که دارای بانک اطلاعاتی متمرکزی برای احراز هویت کاربران هستیم[3,11,12].

شکل(۱) : Single Sign On

۲- انواع SSO

به طور کلی مکانیزم تک ورود در سه حوزه مختلف به کار می‌رود:

۲-۱- Intranet (ESSO[c])

این حالت زمانی اتفاق می‌افتد که کاربر تنها با یک بار لاگین کردن[d]، به سیستم‌های مختلف سازمان دسترسی داشته باشد; در واقع برای کاهش تعداد دفعاتی است که کاربر باید شماره شناسه و کلمه عبور خود را برای ورود به برنامه‌های چندگانه وارد کند.

۲-۲- Extranet (MDSSO[e])

در این حالت کابر می‌تواند تنها با یک بار لاگین کردن، به تمام برنامه‌ها و سیستم‌های همکاران کاری در سازمان‌های مشابه دسترسی داشته باشد; یعنی کاربر می‌تواند به یک سازمان لاگین کند و به منابع بقیه دسترسی داشته باشد زیرا اطلاعات مربوط به احراز هویت و مجوز او، بین دامنه‌های مجاز منتقل می‌شود.

۲-۳- Internet (WSSO[f])

در این حالت مبنای کار بر اساس مرورگرهاست و می‌توان تنها با یک لاگین، به برنامه‌های توزیع شده بر روی سرورهای مختلف دسترسی داشت[1].

در واقع در سطح وب از مکانیزم WSSO استفاده می‌شود که برای حفظ و نگهداری اعتبارنامه از کوکی استفاده می‌کند اما کوکی قادر به انتقال بین دامنه‌های DNS نیست و این یعنی اینکه اطلاعات یک کوکی از یک دامنه در دسترس دیگر دامنه‌ها نیست از این رو MDSSO ارائه شد تا این اطلاعات را بین دامنه-ها انتقال دهد اما چون شرکای کاری دارای محیط‌های ناهمگی هستند پس از پروتکل SAML در آن استفاده می‌شود تا مشکل انتقال اطلاعات کاربر از یک سرور وب به دیگر دامنه‌ها را برطرف کند[2].

۳- ساختار SSO

برای پیاده‌سازی SSO در سیستم‌های مختلف، می‌توان از مکانیزم‌های مختلفی استفاده کرد.

۳-۱- مبتنی بر نشانه

در صورتی که یک مجموعه از اعتبارنامه[g] داشته باشیم می-توانیم از نشانه یا token برای پیاده‌سازی آن استفاده کنیم. در این حالت پس از آنکه کاربر برای ورود به سیستم اقدام کرد، اعتبارش از طریق پایگاه داده حاوی اعتبارنامه‌ها بررسی می‌شود و در صورت تطبیق، یک نشانه[h] به او تخصیص داده می‌شود که برای دسترسی به منابع از آن استفاده می‌کند.

این ساختار را می‌توان در محیط http و با استفاده از کوکی-ها پیاده سازی کرد. کوکی پیامی است که سرویس دهنده به مرورگر می‌فرستد و مرورگر آن را ذخیره می‌کند. پس از آن، هر زمان که مستند وبی از آن سرویس دهنده را باز می‌کند این پیام به آن سرویس دهنده بر می‌گردد تا از این طریق، سرویس دهنده، کاربر قبلی خود را بشناسد و تنظیماتی که او قبلا بنا بر سلیقه‌اش اعمال کرده را دوباره برایش اجرا کند. (مثلا تنظیمات زبان)

در این حالت نیاز به هیچ‌گونه تنظیمات اضافی در سیستم نیست و همین مسئله تفاوت آن را با پروتکل کربروس نشان می-دهد زیرا در کربروس از RPC[i] برای انتقال بلیط استفاده می‌شود در حالی که در اینجا از کوکی استفاده می‌شود[1].

۳-۲- مبتنی بر کلید عمومی[j]

در این حالت سرویس دهنده‌ها و کاربران از طریق جفت کلید مربوطه احراز هویت می‌شوند یعنی با کلید عمومی رمزنگاری و با کلید خصوصی رمز گشایی صورت می‌گیرد و از این طریق یکدیگر را احراز هویت می‌کنند.

۳-۳- همگام سازی اعتبارنامه

در این حالت، مجموعه اعتبارنامه‌های مورد نیاز برای سیستم‌های مختلف، درون یک مجموعه منفرد از اعتبارنامه‌ها ماسک[k] می‌شود و درخواست تغییر اعتبارنامه‌ها به صورت اتوماتیک برای همه سرویس دهنده‌ها ارسال می‌شود.

۳-۴- محل ذخیره اعتبارنامه

اعتبارنامه کاربر می‌تواند در سمت سرویس‌گیرنده[l] یا سرویس‌دهنده[m] ذخیره شود که در صورت ذخیره آن در سمت سرویس‌دهنده در پوشه‌ای به نام Vaults ذخیره می‌شود که حاوی هر نوع اطلاعات حساس کاربر مثل نام کاربری و کلمات عبور و حتی گواهینامه[n] و نشانه اوست. در این صورت کاربران می‌توانند به صورت ایمن و خودکار به وب‌سایت‌ها و سیستم‌های درون شبکه وارد شوند بدون آنکه نیازی به خاطر سپردن اعتبارنامه‌های مختلف داشته باشند.

اما در صورت ذخیره اعتبارنامه در سمت سرویس‌دهنده، نیاز به یک سرویس‌دهنده مرکزی است که اطلاعات را در خود نگه دارد و هر زمان که نیاز باشد، به برنامه‌های مورد نظر ارسال کند[1].

۴- پروتکل‌های SSO

برای پیاده‌سازی SSO از پروتکل‌های مختلفی استفاده می‌شود که هر یک دارای نقاط ضعف و قوتی نسبت به دیگری هستند و بر حسب نیاز از آن‌ها استفاده می‌شود.

۴-۱- پروتکل احراز هویت کربروس

کربروس یک پروتکل توزیع شده مبتنی بر نشانه است که به منظور احراز هویت افراد در شبکه به کار می‌رود. این پروتکل یکی از مهمترین پروتکل‌ها در این زمینه است که نسخه‌های چهارم و پنجم آن، امروزه به طور گسترده به کار می‌روند.

پروتکل کربروس، برگرفته از اساطیر یونان باستان است که به معنای سگ سه سر می‌باشد که این سه سر، نماد سه مولفه AS[o]، TGS[p]، S[q] است که در آن سرویس‌گیرنده پس از تایید هویت توسط AS، کلیدی برای برقراری یک ارتباط امن با TGS دریافت می‌کند و پس از برقراری ارتباط، با درخواست دسترسی به منبع S، یک بلیط دریافت می‌کند و از طریق این بلیط یک ارتباط امن با S برقرار می‌شود. در این روش، کاربر تنها یک بار از اعتبارنامه‌اش به منظور تایید هویت در برابر AS استفاده می‌کند ولی بقیه مراحل چندین بار تکرار می‌شود.

یکی از حملات مهم در این حوزه، حمله تکرار[r]، نام دارد؛ این حمله زمانی رخ می‌دهد که نفوذگر با استفاده از ابزار استراق سمع[s]، بسته‌های اطلاعاتی را از روی سیم برداشته و اطلاعات مهم آنها را سرقت می‌کند اما در پروتکل کربروس امکان وقوع این نوع حملات کاهش می‌یابد زیرا تقاضاهای ارسالی به سرویس دهنده صدور بلیط و سرویس دهنده نهایی، دارای مهر زمان[t] هستند[4].

یکی دیگر از انواع حملات، حمله رمز عبور[u]، نام دارد که برای جلوگیری از وقوع آن، می‌توان از سیستم رمزنگاری کلید عمومی و همچنین رمز ورود پویا استفاده کرد. مشکل دیگر ذخیره شدن رمز ورود در سمت سرویس‌گیرنده است که این مشکل را نیز می‌توان با استفاده از تکنولوژی کارت‌های هوشمند که شامل یک تراشه سیلیکونی جاسازی شده روی یک قطعه کارت پلاستیکی با ابعاد استاندارد است، حل کرد. این نوع تراشه‌های حافظه، قابلیت ذخیره سازی داده‌هایی همچون مشخصات فردی، رمزهای عبور و کلید عمومی را داراست.

روش دیگر برای جلوگیری از وقوع حملات تکرار، استفاده از تکنیک رمز ورود یک بار مصرف[v] (OTP) است. اکثر روش‌هایی که برای پیاده سازی SSO مطرح می‌شوند پر هزینه و زمانبرند چراکه نیاز به تغییرات عمده در ساختار سیستم‌های موجود دارند اما در این شیوه، پس از ورود کاربر به سایت، اطلاعات او مانند رمز ورودش به زیربرنامه‌های دیگر ارسال می‌شود؛ در واقع در صورتی صفحه خوش‌آمدگویی ظاهر می‌شود که کاربر کلمه عبور درستی را وارد کند و اعتبارنامه او طبق پایگاه داده موجود از نام‌های کاربری و کلمات عبور تایید شود، بنابر این می‌توان آن را جایگزین خوبی برای رمزهای ورود ایستا دانست اما به دلیل ریسک بالای ناشی از ارسال اطلاعات خصوصی کاربر در شبکه، باید در انتخاب آن دقت کرد[3].

۴-۲- SAML

روش دیگر بر پایه SAML[w] است که استاندارد بازی برای تبادل اعتبارنامه‌های امنیتی بین شرکای تجاری است که در سیستم‌های مختلفی از جمله Google به کار می‌رود.

SAML استانداردی بر مبنای XML است که برای تبادل داده لازم برای احراز هویت و مجوز دسترسی کاربر بین دامنه‌های امن مختلف به کار می‌رود، یعنی پس از آنکه کاربر از طریق ارائه دهنده سرویس احراز هویت[x]، احراز هویت شد اطلاعات

مربوط به آن مانند جزئیات احراز هویت، نحوه دسترسی و مجوز کاربر در قالب یک پیام XML به بخش‌های دیگر ارسال می‌شود[1,8].

OpenID -۳-۴

شناسه باز[y]، یک سامانه SSO است که یک پروتکل احراز هویت غیر متمرکز به حساب می‌آید و به همین دلیل هر وبگاه دیگری می‌تواند از نرم‌افزارهای آن، به عنوان یک روش برای ورود کاربران خود استفاده کند و چون در مالکیت هیچ کس نیست لذا هرکس می‌تواند یک کاربر شناسه باز و یا حتی فراهم کننده آن باشد. در واقع با پرکردن یک فرم ثبت نام و ایجاد یک کلمه عبور می‌توان به یک کاربر شناسه باز تبدیل شد زیرا ساختار آن، به شکل یک url است[8,9]. اما این سیستم از نظر امنیتی، هنوز دارای مشکلاتی است که باعث می‌شود کاربران در انتخاب آن دچار تردید شوند. به عنوان مثال، فراهم کننده هویت، قادر به دنبال کردن و ردیابی وب سایت‌هایی است که کاربر به آنها وارد شده و این مسئله کاربران را نگران می‌کند[10].

BrowserID -۴-۴

شناسه مرورگر[z]، مشابه شناسه باز است که یک پروژه آزمایشی ارائه شده از سوی موزیلاست و با استفاده از آن می‌توان با یک آدرس ایمیل و رمز عبور به هر سایتی وارد شد اما محدود به فایرفاکس نمی‌شود و دیگر مرورگرهای مدرن نیز در تلاشند تا از این قابلیت بهره مند شوند.

موزیلا ادعا می‌کند که این روش از امنیت بسیار بالایی برخوردار است زیرا شناسه مرورگر، هیچ جزئیاتی درباره اینکه ما از چه وب سایت‌هایی بازدید می‌کنیم را در اختیار سرویس دهنده های دیگر و از جمله سرویس‌دهنده خودش قرار نمی‌دهد[7].

۵- چالش‌های موفقیت SSO

به دلیل وجود زیرسازمان‌های مختلف که در آن‌ها از SSO استفاده می‌شود، انتخاب بهترین روش ایجاد SSO که با تمامی نیازهای آن‌ها مطابقت کند تقریبا غیر ممکن است. همچنین پیاده سازی SSO زمانی موفقیت آمیز است که بتواند نیازهای در حال تغییر سازمان‌های بزرگ را برآورده کند، همچنین با کاربران راه دور سازگار باشد و آن‌ها بتوانند به سهولت به اعتبارنامه‌هایشان دسترسی داشته باشند و آن را به روز کنند. این روش باید امکان مدیریت راحت، استفاده سریع و قابلیت دسترسی بالایی داشته باشد تا کاربران بتوانند به سهولت از آن استفاده کنند و کنترل و حفظ آن برای کاربر و مدیر آسان باشد و از استانداردها و معماری باز استفاده کند تا برای کاربران به صورت یکپارچه، امن و مقرون به صرفه باشد و هزینه مالکیت را کاهش دهد. همچنین قابلیت افزودن کارت‌های هوشمند، توکن‌ها و بیومتریک‌های[aa] احراز هویت مثل اثر انگشت یا ترکیبی از آن‌ها را دارا باشد[2,6].

نتیجه

با افزایش تعداد کاربران و گسترش سازمان‌ها، کاربران و مدیران با معضل تعدد نام کاربری و کلمات عبور مواجه می‌شوند که مدیریت و کنترل آن کار دشواری است اما با استفاده از مکانیزم SSO می‌توان بر این مشکل چیره شد.

روش‌ها و پروتکل‌های مختلفی برای پیاده‌سازی SSO وجود دارد که هر کدام دارای نقاط قوت و ضعف بسیاری هستند و مسئولان باید بر اساس نحوه توسعه و مدیریت منابع، بهترین مکانیزم را انتخاب کنند وگرنه با مشکلات متعددی از جمله آسیب پذیری و نقص در امنیت اطلاعات روبرو می‌شوند.

مراجع

[1] Radha, V., Hitha Reddy, D., "A Survey on Single Sign On Techniques", *Procedia Technology, ELSEVIER*, 2012.
[2] Karunanithi, D., Kiruthika, B., "Single Sign On and Single Log Out in Identity Management", *IEEE*, 2011.
[3] Tiwari, p., Joshi, sh., "Single Sign-on with One Time Password", *IEEE*, 2009.
[4] Jian, y., "An Improved Scheme of Single Sign On Protocol", *IEEE, 2009 Fifth International Conference on Information Assurance and Security*, 2009.
[5] Kiran, L., Sood, S., Singh, K., "A Single Sign On Model for Web Services based on Password Scheme", *2009 First International Conference on Computational Intelligence, Comunications Systems and Networks, IEEE*, 2009.
[6] Mansfield, S., "Single Sign On: matching convenience with security", *Biometric Technology Today, ELSEVIER*, Volume 2011, PP. 7–11, July–August 2011.
[7] www.Certcc.ir.
[8] Wikipedia website: www.wikipedia.com.
[9] OpenId website: http://openid.net.
[10] Learning the OpenID problems, May 14, 2008, http://mateusz.loskot.net/2008/05/14/learning-the-openid-problems.
[11] Information Security Intelligence, September 2011, http://palpapers.plynt.com/issues/2011Sep/sso-flaws.
[12] The Open Group, 2010, http://www.opengroup.org/security/sso.

[a] Kerberos Protocol
[b] Authentication
[c] Enterprise SSO
[d] Log in
[e] Multi Domain SSO
[f] Web SSO
[g] Credential
[h] Token
[i] Remote Procedure Call
[j] Public Key Infrastructure
[k] Mask
[l] Client Side Credential Caching
[m] Server Side Credential Caching
[n] Certificate
[o] Authentication Server
[p] Ticket Granting Server
[q] Application Server
[r] Replay Attack
[s] Sniffer
[t] Time Stamp
[u] Password Attack
[v] One Time Password
[w] Security Assertion Markup Language
[x] Identity Provider
[y] OpenId
[z] BrowserId
[aa] Biometric