הזדהות אוטומטית בין מערכות מידע

אנדרי לבובסקי

<u>מבוא</u>

הצורך בהזדהות אוטומטית בין המערכות הינו טבעי. לדוגמא , מערכת א׳ מבצעת טרנזקציה כלשהי וזקוקה לנתון/ים ממערכת אחרת (מרוחקת) בבעלות ארגון אחר.

למעשה לפנינו צורך לזהות את ״המערכת הפונה״ ולאפשר ניהול ההזדהות לאורך זמן חידוש סיסמא וכד׳.

<u>קצת היסטוריה</u>

בראשית שנות ה 2000 טוייטר רצתה לפתח את עצמה באמצעות מתן אפשרות לחברות צד ג׳ להשתמש בשירותים שלה לצרכים שונים. משתמש קצה פשוט נדרש להזדהות לשרות של צד ד׳ באמצעות שם משתמש וסיסמא של טוויור. למעשה פרטי משתמש הופכים להיות גלויים לצדי ג׳. מובן מאליו החיסרון האבטחתי של השיטה הזו.

בשנת 2005 פיתח Brad Fitzpatrik (בעלים של אתר פופולרי LiveJournal) שיטה להעברת הזדהות בין אתרים/ספקי שרות שונים.

השיטה קיבלה שם : **OpenID Authentication** והפכה לסטנדרט עולמי.
OpenIDהיא מסגרת פתוחה וחופשית לזְהוּת מקוונת מונחית משתמשים.[]זוהי למעשה מערכת SSO גלובלית. בשיטה הזו המשתמשים באינטרנט אינם צריכים לזכור הרבה אמצעי זיהוי מסורתיים כמו שם וסיסמה. במקום זאת, הם רק צריכים להיות רשומים לאתר המספק זהות . בזכות העובדה ש OpenID-היא תוכנת קוד פתוח, כל אתר יכול להשתמש ב OpenID-על מנת לספק למשתמשים באתר שירותי הרשמה והתחברות לאתר OpenID . השיטה פותר את הצורך להירשם לאתרים רבים ולזכור את החשבונות מבלי להסתמך על איזושהי ישות יחידה המנהלת ושולטת בזיהוי הזהות הדיגיטלית. נפרט רק כמה ספקי זהות עולמיים **IDENTITY PROVIDER**

- YAHOO
- FACEBOOK
- IBM
- MICROSOFT

בשנת 2006 הומצא פרוטוקול **OAUTH1** ע״י Blaine Cook באמצעותו ניתן היה לספק גישה לשרותי טוויטר ללא צורך בהזדהות מפורשת.

בשנת 2008 אחרי עבודה משותפת של מהנדסי תכנה מחברות GOOGLE ו AOL פורסם פרוטוקול OAUTH**1**

יש לעשות הבחנה בין OpenID לבין OAUTH . הראשון מספק אפשרות זיהוי משתמש קצה באתרים רבים ללא צורך בהזדהות נוספת. השני מאפשר גישה לשירותים (api׳s) .

השחקנים :

CLIENT

RESOURCE OWNER - בעל המשאב / שרות אותו ה client חפץ לצרוך

RESOURCE SERVER - שרת בו ממקומם משאב / /שרות

AUTHORIZATION SERVER - שרת דרכו מתבצע הזיהוי

AUTORIZATION TOKEN - טוקן (מחזורת) שמאפשר גישה לשרות מוגן

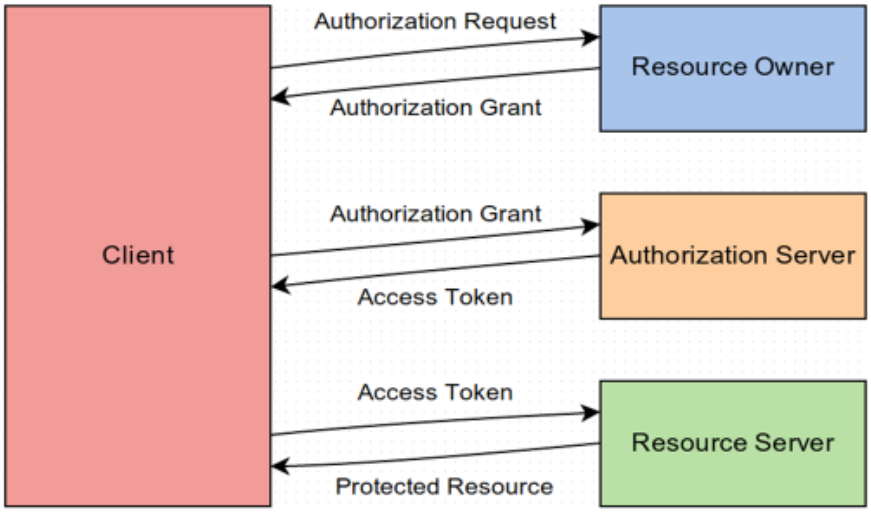
נתאר בקצרה את התהליך :

ה clinet שחפץ לצורך משאב שרות ניגש / פונה אל RESOURCE SERVER עם בקשת הזדהות לשימוש בשרות מוגן כלשהו. בשלב הזה ה client נדרש לבצע הזדהות. במידה וההזדהות צלחה , מוחזר ל client grant code - קוד זמני שבד"כ תקף לפרק זמן מוגבל איתו ה client מבצע פניה נוספת אל authorization server לקבלת AUTHORIZATION TOKEN באמצעותו פונה אל RESOURCE SERVER וצורך את השרות.

הטוקן מוצפן כאמור בפרוטוקול rsa ומבנהו סטנדרטי בתקן

JWT - Jason web token - טוקן סטנדרטי (פרוטוקול RSA)

בשנת 2013 פורסם פרוטוקול OAUTH2 . למעשה זהו פרוטוקול הכי נפוץ לגישה לשירותים b2b.



המנגנון המופעל הוא כדלקמן:

ה client שולח בקשת זיהוי (authorization request). בד"כ בשלב הזה המשתמש מתבקש לבצע הזדהות עם credentials וכתוצאה מכך יקבל authorization grant – קוד זמני איתו הוא אמור לפנות ל authorization server. תהליך דו פאזי נועד לחזק את ההיבט האבטחתי. אורך חיים של authorization grant קצר מאוד.

פניה ל authorization server תחזיר ל client טוקן גישה - access token באמצעותו ה client ייגש לשירותים המוגנים.

אורך חיים של access token ארכיטקטורות שונות משתנה בהתאם לצרכי המערכות ורמת האבטחה הנדרשת. במידה ופג תוקף של access token , קיים בפרוטוקול תהליך החידוש כאשר כתוצאה ממנו נוצר טוקן מחודש - resresh token.