אבטחת מערכות ויישומים ברשת - שיעור 11#

המבחן יהיה אמריקאי, בחירת 33 שאלות מתוך 40. יפורסמו כמה שאלות לדוגמא לקראת המבחן (18/6/2010).

נושא היום: OpenID and Application Vulnerabilities: Un-validated Input:

בהרצאה הקודמת דיברנו על Web SSO, SAML, assertions וריאנו סרטון המדבר על בעיות אותנטיקציה ברשת. כיום נתאר את פרוטוקול ה-OpenID.

OpenID come to solve the identity and profile management problems:

כאשר מישהו רוצה להירשם לאתר מסויים, יתכן ששם המשתמש שרוצה לבחור כבר תפוס, וצריך לנהל שמות משתמשים רבים. כמו כן בהרבה אתרים המשתמש צריך להכניס פרטים – להזין נתונים אלו שוב ושוב בהרבה אתרים. מצב זה שהכל מפוזר בהרבה מאוד אתרים אינו רצוי.

ה-OpenID הוא פרוטוקול המתחיל בזהות המשתמש: מגדירים את זהות המשתמש באמצעות URL המזוהה עם המשתמש, והוא יכול להחזיק כמה URLs שמזוהים איתו. הם מתארים מיהו המשתמש וכיצד לזהות אותו. אלו הם OpenID identity. כאשר משתמש רוצה להזדהות, הוא יתן URL וכך ידעו גם איך לזהות אותו. זה דומה לעולם ה-emails, אך שם יש את הבעיה כאשר מחליפים כתובת אימייל, לעומת כאן שה-identity לא משתנה בכל אופן.

OpenID come to solve the password management problem:

לא רצוי לנהל הרבה סיסמאות בהרבה אתרים ולחילופין גם לא להשתמש באותה סיסמא לכל האתרים (בעיות אבטחה). ב-OpenID המשתמש יכול לנהל כמה OpenID ולהחליט מי ה-OpenID identity provider. יהיה שם משתמש וסיסמא לכל identity provider כזה. תחת הנחה שהמשתמש לא רוצה לקשור את עצמו ליחיד כזה, אלא לכמה, הוא ינהל אותם. זה סביר לנהל ולזכור 4-3 סיסמאות. ה-OpenID providers מנהלים את הזהות והסיסמאות, מולם מזדהה המשתמש.

OpenID Overview:

- חלק מנושא ה-OpenID הוא single sign-on. כאשר משתמש רוצה להזדהות הוא מספק לאתר את ה-OpenID identity. האתר יודע לאתר מי ה-OpenID provider ופונה אליו ומבקש ממנו אימות שמהדפדפן המסויים הזה אכן פונה היישות שמקושרת ל-OpenID הזה.
- ה-identity provider מספק את אימות הזהות ונותן לספק השירות גם נתונים נוספים החשופים לבחירת המשתמש. ה-identity provider עונה לאתר המבקש זיהוי תשובה על השאלה: האם אתה מזהה כרגע את ה-id הזה מהדפדפן הזה כרגע. הוא מחזיר תשובה ב-redirect אל הדפדפן.
- מי שמבצע אימות זהות יהיה ה-OpenID identity provider. אם התבצע כבר זיהוי מולו, לא יתבקש המשתמש לזיהוי נוסף. אחרת, יתבקש זיהוי.

OpenID Highlights:

- OpenID הוא URI – יעד ברשת, שיכול לכלול גם אינפורמציה אודות פרופיל המשתמש – כך המשתמש לא צריך להזין כל פעם מחדש את האינפורמציה לגביו, ויש ביכולתו האפשרות לקבוע מה מהאינפורמציה שלו תהיה חשופה כלפי חוץ.
- זהות זו יכולה לשמש גם לאותנטיקציה וגם להעברת attributes, אך בניגוד ל-SAML, אין כאן התייחסות ל-authorization.

De-Centralised:

- מנגנון זה של ה-OpenID הוא מבוזר. אין SSO יחיד עבור המשתמש. ב-SAML האתרים הלוויניים מחליטים מי ה-idp שלהם, ודורשים מהמשתמש לעבוד מולו אם הוא רוצה שירות מהם. כאן המשתמש יכול להחזיק יותר מזהות אחת מ-idp שונים וכך לתת יותר בחירה וכוח בידי המשתמש. כך המשתמש יכול להחליט מתוך אלו שבהם הוא בוטח.
- אין כאן הנחה על סביבת העבודה, אלא הנחת יכולות סטנדרטיות על הדפדפן – בדומה ל-SAML.
- משתמש לא רק יכול ליצור ID בספקים שונים, אלא אפילו יכול לבנות OpenID provider server בעצמו. השאלה היא האם אתר אליו פונה יכול לסמוך עליו. זה מעלה שאלת ביטחון בין ספק ה-ID לספק השירות.

ישנה רשימה גדולה מאוד של OpenID providers שקיימים היום (AOL, Google,...).

ה-OpenID עונה על שאלת ה-authentication אך לא עונה על שאלת ה-authorization.

OpenID terminology:

- OpenID provider: מספק את ה-assretions.
- Relying Party: ספק השירות שמבקש זיהוי מה-IP למשתמש.

OpenID Login Process:

- המשתמש פונה ל-RP והוא פונה ל-IP ושואל האם הדפדפן הזה מתאים אליו והוא הזדהה מול ה-IP.
- ה-IP שולח את הזהות ואולי גם attributes לפי הגדרת המשתמש אל ה-RP. הבעיה כאן היא שה-token עובר דרך ה-browser ויכול להשתנות ע"י המשתמש, לכן יש צורך באימות ה-token באחת משתי האופציות:
  - חתימה דיגיטלית סימטרית / אסימטרית של ה-token ע"י ה-IP, והיא תבדק ע"י ה-RP.
  - העברת ה-token אל ה-IP לבדיקת אמינותו (סוג של לבקש מחדש את ה-token).

OpenID authentication request:

- Checkid setup: ה-IP מקפיץ חלון הזדהות למשתמש.
- Checkid immediate: ה-RP שואל את ה-IP האם למשתמש הזה יש כרגע סשן פתוח מולו – ומקבל תשובה כן או לא, בניגוד לקודם שם מבקש ה-RP מה-IP לזהות את המשתמש. כלומר כאן הבקשה לא להקפיץ חלון login אלא רק לדעת האם המשתמש מחובר כרגע.

ההבדלים בין אלו נועדו לשלוט בחווית המשתמש – האם יוקפץ או לא יוקפץ חלון login למשתמש.

Authentication response verification:

- Check authentication: ה-RP מקבל מהדפדפן של המשתמש id token, ומבקש מה-IP תשובה לשאלה האם ה-token הזה אמין או לא.
- Associate: נועד ליצור את אותו shared secret סימטרי או אסימטרי. ב-associate מחליפים ה-RP וה-IP shared secret, וכשיש כזה (למשל סימטרי), אז יכולים להשתמש בחתימה דיגיטלית סימטרית על ה-token באמצעות hash קריפטוגרפי כלשהו (פירוט במצגת). ה-shared secret הזה הוא עבור פרק זמן כלשהו, ולא צריך לבצע check authentication כל הזמן, כי ה-RP יכול לבדוק בעצמו את כל בקשות ההזדהות. אחת לכמה זמן יוגדר associate חדש. כמובן שלאורך זמן ובשימוש מאסיבי שיטה זו יעילה יותר מהקודמת.

User authentication to OpenID identity Provider (IP):

...

OpenID Advanteges:

- SSO.
- לא צריך לנהל הרבה סיסמאות, רק כמה מול ה-IP שבחרנו לעבוד מולם.
- לא צריך למלא כל פעם את האינפורמציה האישית מחדש, כל עוד מה שהוגדר כ-public פתוח לספקי שירות.
- ישנה זהות גלובאלית שכל אתר המוכן לסמוך על ה-IP יכול להשתמש, ומצד שני זה מבוזר – אין IP יחיד בעולם, אלא כמה שכל משתמש יבחר מתוכם 2-3 איתם רוצה לעבוד.
- הפרוטוקול בנוי על סטנדרט מסויים שאינם משוייכים לאף חברה או קבוצה.

אין סיכום משעה שניה.