**דו"ח סופי למיני פרויקט באבטחת מחשבים**

**רון יהודה ואורין זינגר**

**יוני 2017**

**הסבר על הפרויקט:**

הפרויקט נותן פתרון איכותי ונוח למשתמשי קצה לשמירת הסיסמות שלהם בשרת מרוחק באופן מאובטח אך שקוף למשתמשים על ידי מספר רב של הגנות. במימושנו, לא השרת או המאזין למדים דבר מן ההתקשרות כך שפרטיות המשתמשים נשמרת.

**צד לקוח:**

תוסף לכרום הנותן ללקוח להירשם לשירות, להתחבר אליו, לייבא את הסיסמות השמורות שלו ולהוסיף סיסמאות חדשות.

בדף האפשרויות של התוסף ישנן 5 אפשרויות מרכזיות:

1. Signup- ההרשמה מבצעת ולידציה מול השרת על שם המשתמש שהוא לא קיים במערכת כבר. אם הוא קיים נשלחת הודעה בהתאם. אחרת, פרטי המשתמש מוספים לשרת ונוצר קובץ סיסמות עבור הלקוח.
2. Login- המשתמש שולח בקשת התחברות לשרת אשר במידה והפרטים נכונים, מייבאת את קובץ הסיסמות ללקוח. התוסף נכנס למצבlogged in שבו הממשק משתנה ומופיע כפתור logout במקום כפתור ה-login ופרטי המשתמש נשמרים מקומית לשימוש עתידי.
3. Manage Passwords- בלחיצה על הכפתור תפתח למשתמש טבלה המציגה את כל כתובות האתרים והסיסמות השמורות עבור אותו אתר, וניתן למחוק דרכה.
4. Logout- כפתור המעדכן את השרת בקובץ הסיסמות המקומי ומוחק את כל המידע המקומי. בנוסף הוא מחזיר את התוסף למצב המקורי- בו המשתמש לא היה מחובר. העדכון בלחיצה על הכפתור היא אינה היחידה- בהמשך נפרט על עדכון השרת בכל עדכון של סיסמה חדשה.
5. Delete account- ברגע שהמשתמש לא מעוניין שהפרטים שלו ישמרו על השרת הוא ימלא את פרטיו (אימייל וסיסמה) וילחץ על הכפתור Delete account. לאחר הודעת ווידוא שהוא אכן מעוניין למחוק את החשבון שלו תשלח הודעה לשרת לבצע זאת. השרת ימחק את קובץ הסיסמות שלו ואת הפרטים שלו ממסד הנתונים

על שדות המשתמש מבוצעת בדיקות של אימייל תקין וסיסמה המכילה לפחות 12 תווים.

אם יש השרת אינו זמין או שהניסיון להתחבר נכשל אז נשלחת הודעה מתאימה למשתמש.

**צד שרת:**

בזמן הפיתוח בחרנו להשתמש בחבילת WAMP העונה לדרישות הback-end וכן העלאה פשוטה לweb hosting

הכוללת: שרת Apache, מסד נתוניםMySQL וכמובן PHP.

את הגרסה הסופית העלו העלנו ל http://ronyehbgu.000webhostapp.com/bgupass והתוסף מוגדר לעבוד עם השרת המרוחק.

כל בקשת HTTP POST עוברת בדיקה למניעת XSS ולמניעת SQL Injection.

קובץ הסיסמאות נשלח מוצפן בפורמט JSON לאחר שהמשתמש זוהה ואושר על ידי השרת.

דף הlogin.php מאזין לבקשות התחברות , דף הsignup.php לבקשות הרשמה ודף הser.php לבקשות עידכון.

בסיס הנתונים ברשת מכיל שדות שם משתמש(אימייל), סיסמת התחברות, מלח וחתימת האוטנטיקציה.  
קבצי ססמאות המשתמשים נשמרים תחת תיקיית userspass בשרת. נושא האבטחה מפורט בהמשך.

**Workflow:**

לאחרlog in מוצלח, icon התוסף עובר מצבע בהיר לצבע כהה המעיד על מצב פעיל בו קובץ הסיסמות נגיש וניתן לשמור או לקבל סיסמות ממנו.

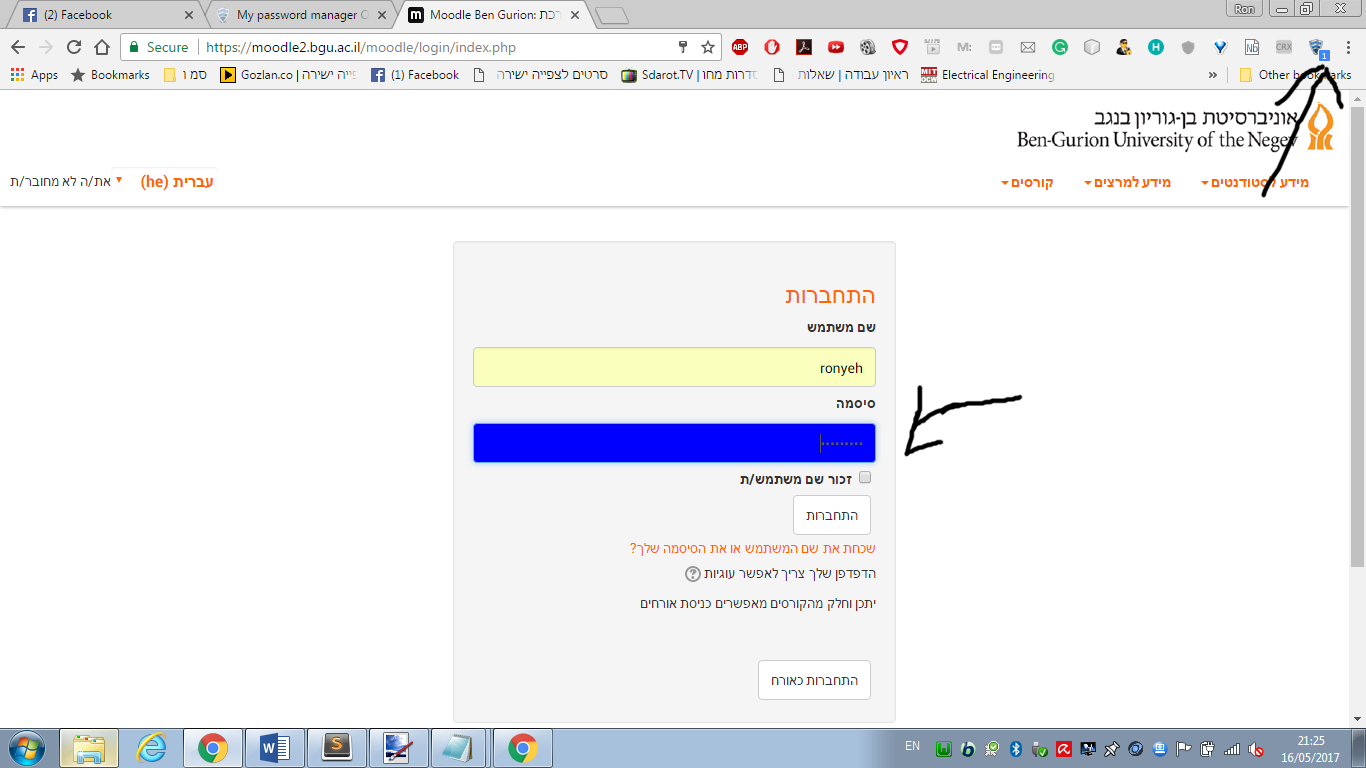
כאשר התוסף במצבlogged in ובעת גלישה בדף אשר יש בו form:

אם המשתמש שולח את הform יחד עם סיסמה חדשה – נפתחת הודעה למשתמש כדי לדעת האם ליצור סיסמה חדשה עבור הכתובת הנוכחית.

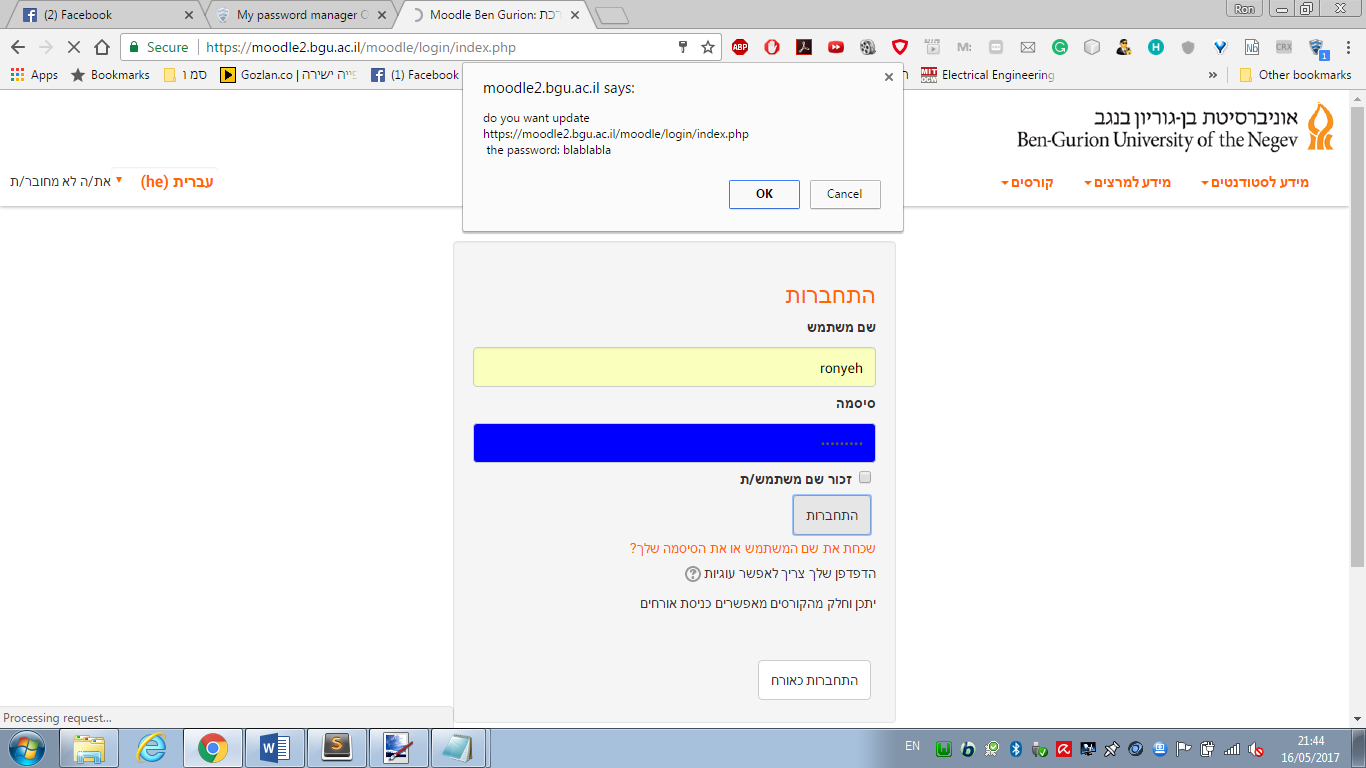
אחרת, אם קיימת סיסמה כבר לכתובת הנוכחית והמשתמש הכניס סיסמה שונה אז הודעה נפתחת למשתמש כדי לדעת האם לעדכן סיסמה קיימת.

במידה והמשתמש מבקר בדף שבו נשמרה סיסמה עבור הכתובת הנוכחית:

בicon של התוסף מקבל notification על סיסמה זמינה בעמוד זה. בעת לחיצה על שדה הסיסמה בדף, רקע השדה משתנה לכחול ומופיעה הסיסמה השמורה.



בכל עדכון של סיסמה או יצירת סיסמה חדשה, היא תשמר מקומית ובנוסף יישלח לשרת קובץ מעודכן של הסיסמות (בנוסף עם פרטי המשתמש לזיהוי וחתימת הקובץ העדכני).



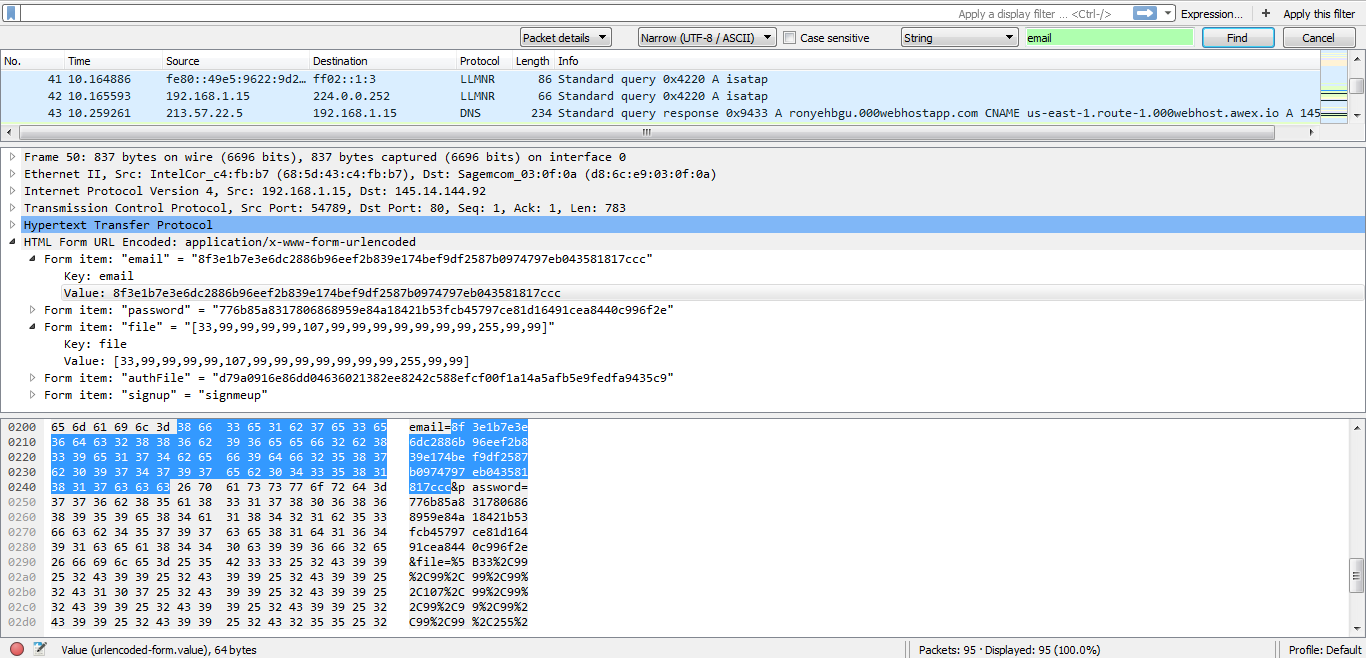
**אבטחת התוכנה מחולקת למספר רבדים:**

1. המשתמש נרשם לשירות עם אימייל וסיסמה, הנתונים עוברים ולידציה כגון אורך הסיסמה הינו 12 ומעלה, תקינות שם המשתמש (בחרנו עפ"י אימייל) ובנוסף, יצרנו מנגנון המחשב (עפ"י אורך, שונות, נוכחות אותיות גדולות ומספרים) את חוזק הסיסמה ונותן לה קרדיט על מנת להמליץ למשתמש לחזק את סיסמתו אם צריך.
2. מהסיסמה שהמשתמש בחר להרשמה נגזרים 3 סודות חדשים על ידי שימוש ב HMAC (מגובבת ב256-SHA) עם 3 מפתחות שונים.
   1. סוד הרשמה- אותו שולחים יחד עם האימייל על מנת להירשם/להתחבר לשירות.
   2. סוד הצפנה- בה משתמשים על מנת להצפין ולפענח את קובץ הסיסמות.
   3. סוד האוטנטיקציה- בה משתמשים על מנת לוודא שהקובץ ששלחנו לשרת אכן הקובץ שאותו אנו  
       מצפים לקבל בחזרה.

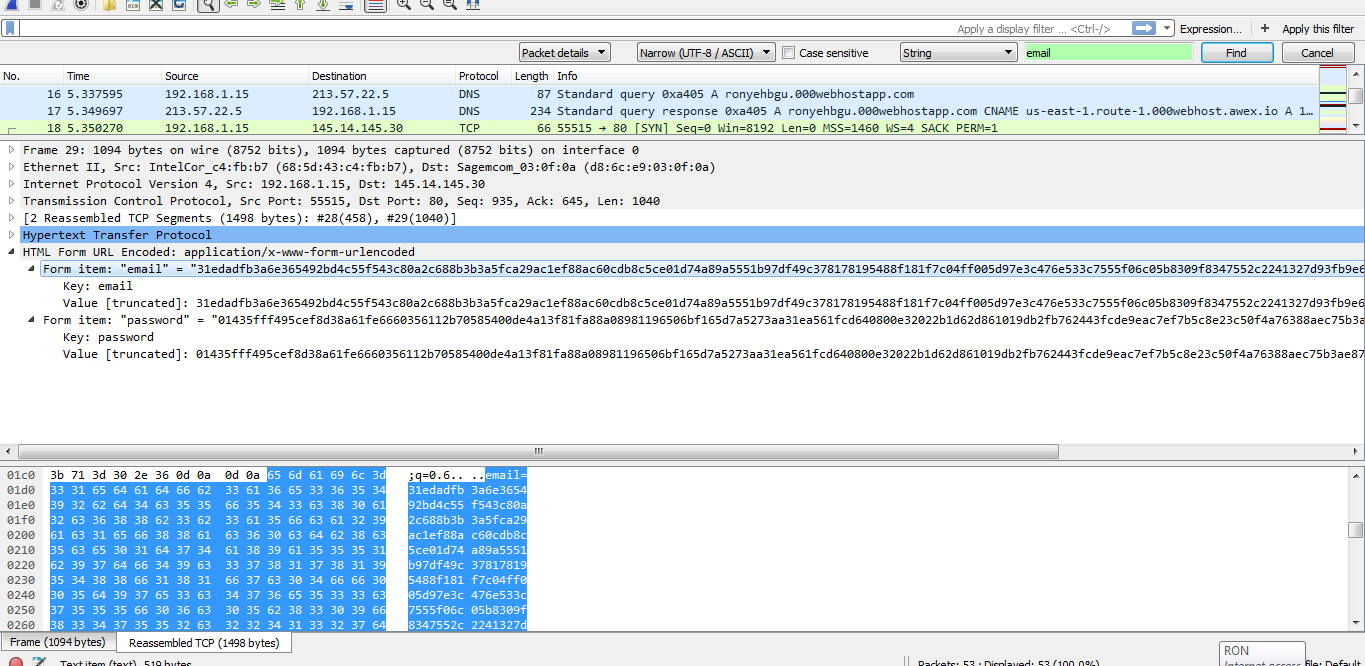
מצאנו לנכון להשתמש בHMAC ולא על ידי פעולת גיבוב SHA-256 בלבד על קלטים שונים (אך דומים) מכיוון שפרוטוקול יצירת הסודות ידוע ואנחנו מעוניינים להקשות על מתקפת Rainbow-tables מפני מאזין או MITM לגלות את הסיסמה הראשית או שם המשתמש.

1. את הסוד שהשרת מקבל הוא שומר באופן ממולח- גיבוב של ערך רנדומלי (המלח) שמשורשר לסוד שהגיע. הגיבוב מבוצע על ידי SHA-256. שם המשתמש נשלח מגובב וכך נשמר בבסיס הנתונים.  
   עבור כל לקוח נוצרת תיקייה תחת userspass שבשרת. התיקייה מקושרת לשם המשתמש המרובב ובה מאוחסן קובץ הססמאות המוצפן שלו.
2. לאחר התחברות לשרת, קובץ הסיסמות וחתימתו נשלחים למשתמש. לפני פיענוחו, יש בדיקת אימות שהקובץ הינו הקובץ המקורי שנשלח. אם קובץ הסיסמות נפגע אז תוצג הודעה למשתמש בהתאם. באופן דומה גם עבור מקרה שהקובץ נמחק מהשרת.
3. בכל הוספה או עדכון של סיסמה ע"י המשתמש, מתעדכן קובץ הסיסמות המקומי וחתימת האוטנטיקציה. הם נשלחים לשרת (ביחד עם פרטי המשתמש שהוזנו בעת ההתחברות לאימות מבקש העדכון). אם השרת אינו זמין אז תוצג הודעה למשתמש וקובץ הסיסמות ישמר מקומית בלבד עד שהשרת יחזור להיות זמין. לכן בחרנו שכל logout ילווה גם בflush של הקובץ המקומי לשרת (בנוסף למחיקת כל הפרטים המקומיים).
4. בנוסף, מעבר לדרישות הפרויקט, במקום שהאימייל וסוד ההתחברות ישלחו אל השרת כplain text (מגובבים) השתמשנו באלגוריתם ההצפנה RSA.   
   בכל login לשרת, הלקוח מקבל מודולוס ואקספוננט ציבורי של מפתח פרטי הנמצא בשרת, שומר אצלו ומצפין את שם המשתמש והסיסמה, ולמעשה כל התקשרות נוספת תעשה על פי הערכים המוצפנים, ותפוענח בשרת.
   1. אנחנו מודעים לעובדה לRSA יש תקורות גבוהות יחסית ובפועל משתמשים בו להעברת מפתח סימטרי, אך מדובר בכמות מידע קטנה ולצרכים דידקטיים בלבד. בנוסף מכיוון שהמפתח הציבורי הנשלח אינו חתום ע"י CA , הדבר אינו מגן מפני MITM. להתגבר על בעיה זו ישנה אפשרות של הטמעת המפתח הציבורי בלקוח מראש, ובכל התקשרות תבדק ה fingerprint של המפתח הציבורי בכל התקשרות ראשונית שבה הוא נשלח. כדי למנוע replay attack יש להוסיף nonce שישלח מהשרת
   2. בעת ההרשמה בלבד השארנו את פרטי המשתמש להישלח כplain-text ולא תחת RSA כדי שנוכל לראות את ההבדל בין פרטי המשתמש המוצפנים בRSA ששולחים בעת התחברות או עדכון / הוספה של סיסמה לקובץ הסיסמות. מצורפות צילומי פקטות ב Wireshark:

כאן ניתן לראות שבעת ההרשמה, ניתן לראות את פרטי המשתמש שנשלחים לשרת: ערכו המגובב של האימייל ron@orian.com"" עפ"י הדרך בה בחרנו לגבבו: HMAC(username, username +"3"), וסוד ההתחברות הנגזר מ- "123456789012" עפ"י (HMAC (Mpassword, Mpassword + "0".



כעת, ליצירת LOGIN מוצלח וקבלת קובץ הסיסמות אלו הפרטים שנשלחו בפועל לשרת,  
ולמעשה אלו אותם ערכים שנשלחו לשרת בעת ההרשמה לאחר שעברו הצפנה באמצעות RSA.



**הוראות הפעלה:**

לצרכי בדיקה לדף הoptions של הextension ישנו משתנה גלובלי המציין אם ברצון הבודק לעבוד במצב של debug mode אשר באמצעותו ניתן לראות פרטים בconsole בעת שימוש בתוסף.

לדף ה-options וה- background הextension ישנו משתנה גלובלי אשר בו מציינים את כתובת השרת (מקומי או מרוחק).

העלנו את השרת לכתובת <http://ronyehbgu.000webhostapp.com/bgupass/> והתוסף מכוון כברירת מחדל

לעבוד עם השרת המרוחק.

תחת תיקיית השרת ניתן לראות את קובץ בסיס הנתונים, דפי הphp ופרטי מפתח הrsa.

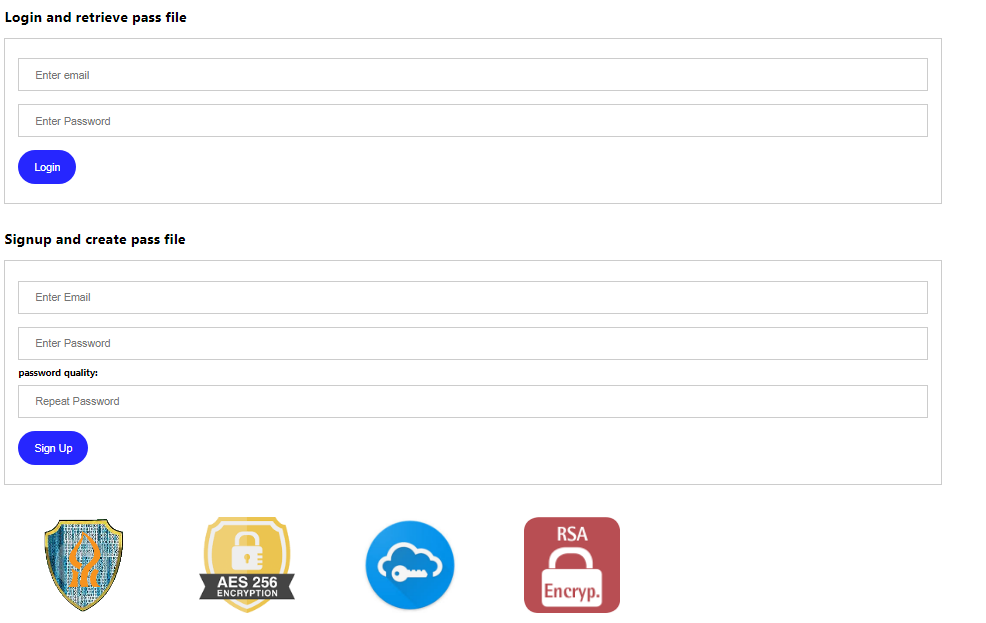
להתחלת השימוש יש לפתוח את כרום תחת מצב מפתח ולבחור בתיקייה ההרחבה ב .load unpacked extension

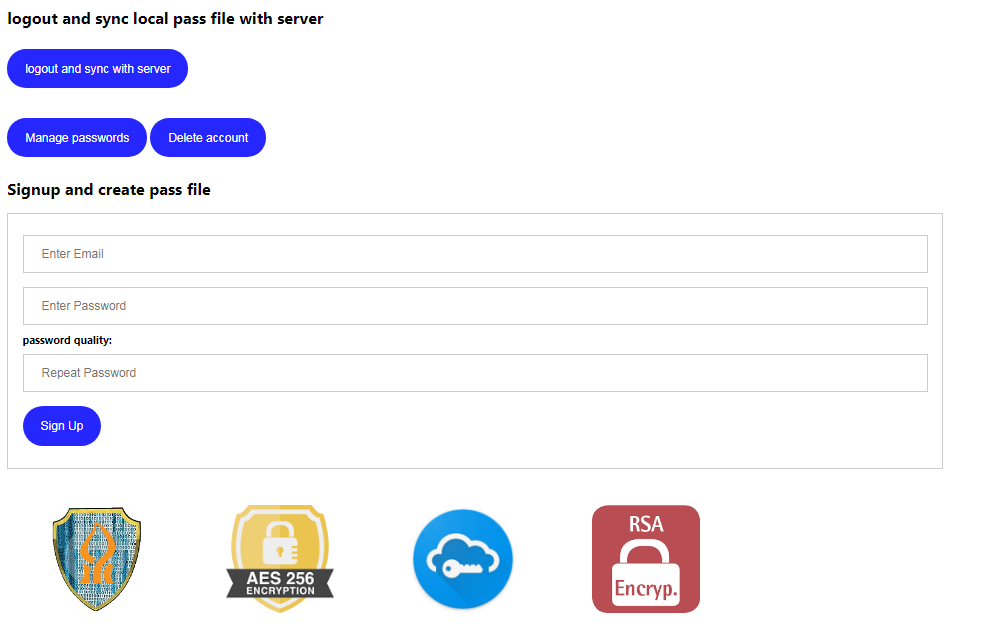
**Documentation:**

|  |  |
| --- | --- |
| RSA | www-cs-students.stanford.edu/~tjw/jsbn/ |
| בניית התוסף | [developer.chrome.com/extensions](https://developer.chrome.com/extensions)  w3schools.com |
| יצירת מפתח RSA פרטי ומוצפן | openssl genrsa -256aes –out private.key 2048 |
| JS-AES | point-at-infinity.org/jsaes/ |
| JS-SHA256 | caligatio.github.com/jsSHA/ |
| JS-HMAC-SHA256 | code.google.com/p/crypto-js |

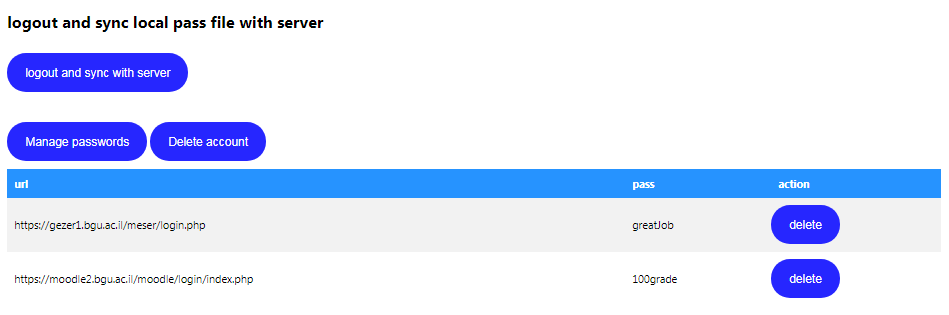
**תמונות להתרשמות**:

לפני התחברות לתוסף:

  
  
  
לאחר התחברות לתוסף:



מנהל הסיסמות:



בשרת , עבור המשתמש שהתיקייה המודגשת שייכת לו, זהו קובץ ססמאותיו:   
  
