

Hard Coded Password in code – not a good practice

1. כל מי שראה את הקוד רואה את הסיסמה
2. הסיסמה דולפת ל Git כשומרים את הקוד
3. אי אפשר להחליף סיסמה בלי ל�מפל את הקוד מחדש
4. סיכון אבטחה חמור לפי OWASP (Open Web Application Security) Project שהוא ארגון בינלאומי ללא מטרות רווח שמתמקד באבטחת אפליקציות תוכנה)

דוגמה לא טובה – סיסמה בקוד

1. string user = "abc@gmail.com";
2. **string pass = "abcd1234";**
3. SmtpClient smtp = new SmtpClient(...);
4. smtp.Credentials = new NetworkCredential(user, pass);
5. **קל מידי לגנוב את הסיסמה ✓**

פתרון 1 - הכנסת סיסמה ב TextBox בצורה מוסתרת

1. המשתמש מכניס סיסמה בזמן ריצה
2. יתרונות: הסיסמה לא נשמרת בשום קובץ וניתן לשנות את הערך בזמן ריצה
ambilijkelijkely מבליל קומפלט מחדש את הקוד
3. חסרונות: המשתמש צריך להקליד כל פעם (מתאים יותר לסיסמת כניסה למערכת)
4. דוגמה:

```
textBoxPass.PasswordChar = '*'; //masking can be done in properties also
```

```
string pass = textBoxPass.Text; //get the password entered by the user
```

פתרונות 2 – משתנה סביבה Variable

1. מוגדר כמשתנה סביבה במערכת הפעלה של המחשב
2. יתרונות: לא נמצא בקוד ולא בGIT וניתן לשנות אותו מבלי ל战ריפל מחדש את הקוד
3. חסרונות: יש צורך להגדיר ידנית בכל מחשב שרצה עליו התוכנה אבל מבחינות אבטחה מומלץ ע"י OWASP

פתרון 2 – משתנה סביבה Variable

הוספה משתנה סביבה חדש:

1. לחץ Start

2. חפש: Edit the system environment variables

3. בחלון → לחץ על: ...Environment Variables

תראה שני אזורים:

User variables • (למשתמש הנוכחי בלבד)

System variables • (לכל המשתמשים — דורש אדרמן)

4. לחץ ...New

5. מלא:

:Variable name •

APP_PASS

:Variable value •

לדוגמה: abcd1234

6. שומר → OK

⚠ לאחר השמירה לפחות פעמים צריך להפעיל מחדש את Visual Studio כדי שהמשנה ייקלט.

פתרון 2 – משתנה סביבה Variable

تعيين משתנה הסביבה לקוד של C#:

```
string pass = Environment.GetEnvironmentVariable("APP_PASS");
if (string.IsNullOrEmpty(pass))
{
    MessageBox.Show("Environment variable APP_PASS is not set!");
}
```

פתרון 3 – App.Config

1. זהקובץ הגדרות (קונפיגורציה) שנשמר ליד האפליקציה (ב bin)
2. יתרונות: ניתן לשנות אותו בצורה נוחה מביי לקמפל מחדש את הקוד
3. חסרון: הסיסמה גלויה אם לא מוצפנת ולכן מסוכן להכניס ל GIT
4. ניתן להכניס ל GIT בלי בעיה אם לא מדובר במידע רגיש
5. ניתן לשמר בו מידע רגיש בתנאי שלא שומרים את הקובץ ב GIT ע"י הוספה App.config לקובץ .gitignore. ואז App.config לא ישמר ב GIT
6. במקרה שהקובץ לא נשמר ב tog כדאי להכניס ל GIT את הקובץ
`<add key="APP_PASS" value="REPLACE_ME" />`
ולעדכן בו את ה `value` הנכון

פתרון 3 App.Config – 3

הוספה Item חדש (key/value) ל App.config
אם הקובץ לא קיים, מוסיףם חדש לפרוייקט:

Right-click project → Add → **New Item** → *Application Configuration File*

לאחר מכן מוסיףם סעיפים appSettings ומוסיףם Item חדש (key/value)

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="APP_PASS" value="abcd1234" />
  </appSettings>
</configuration>
```

פתרון 3 App.Config – 3

טעינה value לkey מה- App.config מהתוך קוד C# :

```
using System.Configuration;  
string pass = ConfigurationManager.AppSettings["APP_PASS"];  
if (string.IsNullOrEmpty(pass))  
{  
    MessageBox.Show("Key 'APP_PASS' not found in App.config");  
}
```

פתרון 4 – שמירת הסיסמה בצורה מוצפנת בתוך קובץ או DB

1. הסיסמה נשמרת מוצפנת בתוך קובץ או DB
2. הcy בטוח מקומית
3. דורש הצפנה ופיענוח. דורש ניהול מפתחות לפעםים
4. נראה בהמשך

השוואה בין השיטות

- .1 TextBox with mask - הći בטוח מצד אחסון אבל לא נוח לשיסמאות ארוכות שאי אפשר לזכור (כמו PASS_App).
- .2 Environment Variable - מאובטח יחסית, כי זה הגדרות פיר מחשב.
- .3 App.config - נוח לשינויים, אבל מסוכן כSSHומרים מייד רגיש (במקרה של מייד רגיש עדיף לא לשמר בGIT).
- .4 Encryption - הcci מאובטח אבל הcci מורכב. נראה בהמשך.

תרגיל

תקו את התרגיל הקודם שבו שמרת APP_PASS בצורת Hard Coded לפי התנאים הבאים:

1. טיענת הסיסמה מהמשתמש ע"י TextBox with masking.
2. במידה והסיסמה לא הוזנה ע"י המשתמש טען אותה ממשנה סביבה בשם APP_PASS.
3. במידה והסיסמה לא קיימת ממשנה סביבה טען אותה מטור קובץ App.config.
4. במידה והסיסמה לא הוזנה בשום שלב, הצג הודעה מתאימה. במידה והצלחת לשלוף את הסיסמה המשר את הקוד כרגע ושלח את המיל.